The Itanium (I64) family

Sima Dezső

Nov. 2018

(Ver. 1.0)

The Itanium (IA-64) family

- 1. The early history of the Itanium (IA-64) family
- 2. The IA-64 ISA underlying the Itanium family
- 3. Overview of the evolution of the Itanium family
- 4. The Itanium line
- 5. The Itanium 2 McKinley/Madison platform
- 6. The Itanium 2 Montecito/Montvale DC platform
- 7. The Boxboro MC platform
  (4-core Tukwila/8-core Poulson/Kittson
- 8. The end of life of the Itanium family
- 9. References

The Itanium family (1)

The Itanium (IA-64) family [69]

1. The early history of the Itanium (IA-64) family

1. The early history of the Itanium (IA-64) family [1], [2], [3]

1.1 Early VLIW developments in HP preceding the HP-Intel cooperation

- EPIC architectures with static dependency resolution have their roots in VLIW designs.

  About 1981 VLIW developments were started and resulted finally in two production machines about 1987, one of them was the Cydra-5 from Cydrome the other the Trace Computer from Multiflow.
- About 1989 both firms went bankrupt.
- In 1989 the co-funder of Cydrome and chief designer of the Cydra-5 VLIW machine (Bob Rau) along with one of the key designers (Mike Schlansker) were hired by HP.
- In 1990 also Josh Fisher, the designer of the first VLIW compiler (ELI-512) at Yale who became one of the key persons of the other manufacturer of commercial VLIW machines (Multiflow Trace line) joined HP after Multiflow ceased operation.
- Fisher convinced HP's management that a VLIW-based machine would outperform a traditional RISC machine [1].
- Accordingly, in 1993 HP decided to develop a new architecture based on VLIW to provide an evolution path to their PA (Precision Architecture) line, called the PA-WW (PA Wide World).

1. The early history of the Itanium (IA-64) family (2)

1.2 The joint HP-Intel investigation about the justification for developing a VLIW based 64-bit architecture [1], [2], [3]

- In 12/1993 HP approaches Intel to jointly develop a 64-bit VLIW based architecture.
- Intel agreed to work jointly with HP on a new VLIW-based 64-bit design.
- During the first half of 1994, these two teams met regularly and exchanged information on 64-bit directions and capabilities.
- To provide protection of mutual intellectual property, the teams agreed to establish a neutral site at an HP training office.

  A conference room was dedicated to the task, with two large filing cabinets, one assigned to each company.

  All of the notes, along with any printed material received from the other firm, were stored in these cabinets and locked when the teams left the room.
- The investigations went well, and on 6 June 1994, HP and Intel announced a joint research activity to develop a 64-bit VLW-based instruction set architecture that would become the basis for a line of Intel microprocessors.

  The PA-WW was used as the starting point for the joint design and John Crawford of Intel became appointed as the leader of the joint team.

1.3 Announcing the first EPIC processors [4], [5]

- Subsequently, over the next year or two the joint team produced a series of specifications, including the ISA that reflected the best view of both companies.

  The joint instruction set development preceded a massive effort at Intel to prepare all of the products needed to successfully launch a new computer architecture.

- In 10/1997, after about three years of joint development work, Intel and HP unveiled the new architecture, dubbed the IA-64 ISA.
- Also the term EPIC (Explicitly Parallel Instruction Computer) was announced to designate the principle of operation of the new ISA.

  In addition, Intel disclosed their schedule to launch EPIC processors.

- Accordingly, Intel slated the first EPIC processor, dubbed Merced for 1999 with a clock speed of 1 GHz.

  A second chip generation was also announced with twice the performance of Merced to be launched in 2001.

1.4 Emergence of the first IA-64 processors [4], [5], [6]

- In fact, both processors were significantly delayed, Merced appeared only in 5/2001 and the second generation EPIC processor, termed McKenly in 7/2002.

  In addition, Merced did not achieve its target speed, the processor was shipped with 800 MHz clock frequency instead of the planned 1 GHz.

  This was embarrassing for Intel as e.g. Intel's own Pentium 4 line reached 1.5 GHz about half a year earlier.

  All in all, the first EPIC processor, called Merced was strongly disappointing.

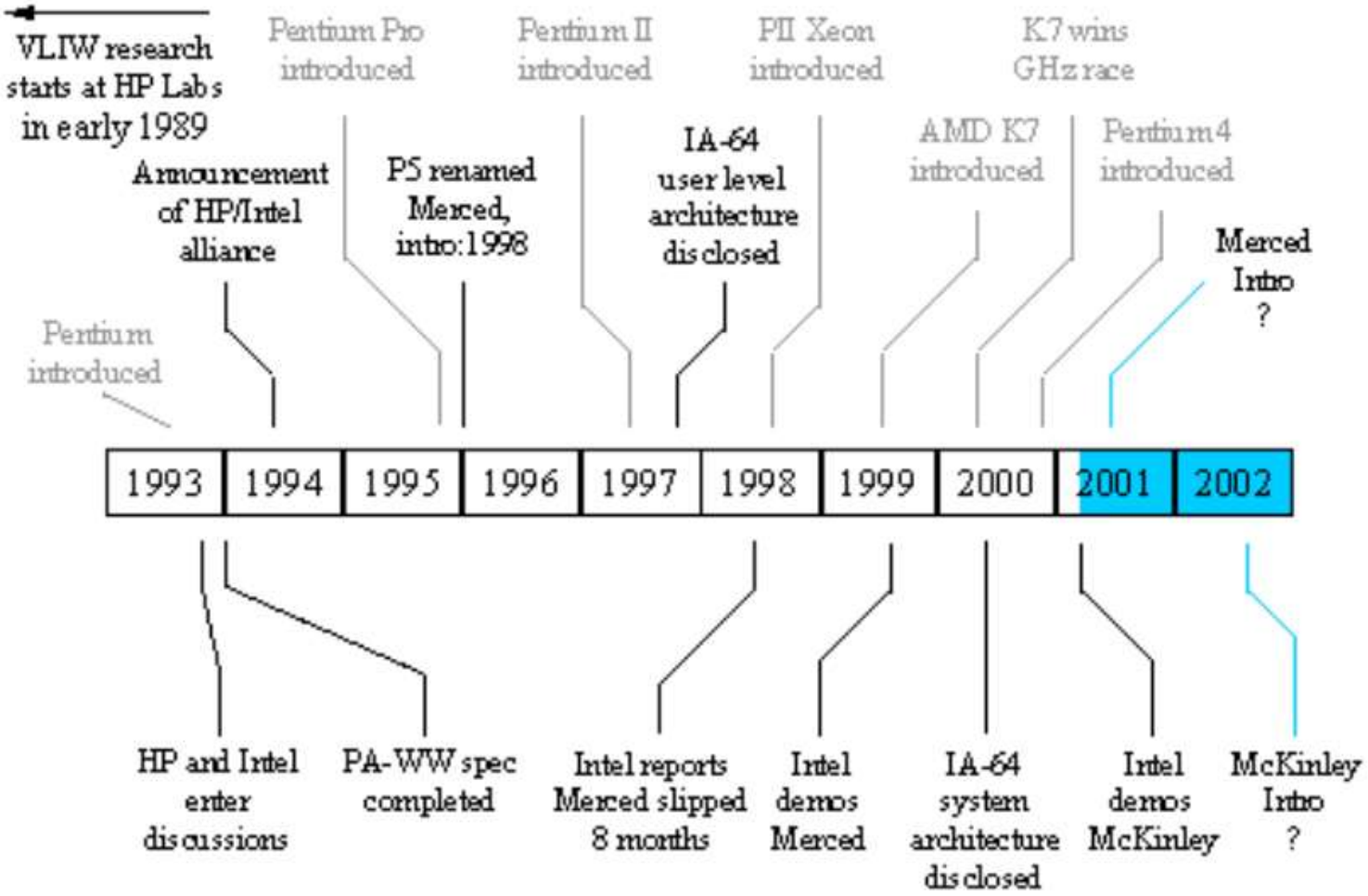
- In 10/1999 Intel announced that the official name of the processor line will be Itanium [6].

  According to Intel officials the name Itanium was chosen to "reflect the strength and performance of the processor".

  Nevertheless, within hours, the name Itanic had been coined and circulated on a Usenet newsgroup, a reference to the ocean liner *Titanic* that sank in 1912 [6].

1. The early history of the Itanium (IA-64) family (5)

The timeline of the early development of EPIC processors [7]



1.5 Market reception of the first IA-64 processors-1 [1]

- Before launching the Itanium line (about 1995-2001) the IA64 team believed that it would penetrate the high end workstations market.
- But over time their target market segment was changed to first the server market and then to high end servers only.

  So the expected total market volume shrunk tremendously from an estimated 200 million units to an expected 200 000 units.

Market reception of the first IA-64 processors-2 [8], [9]

Also industry analysts predicted that IA-64 would dominate in servers, workstations, and high-end desktops, and eventually supplant RISC and CISC processors for all general-purpose applications [8], as indicated in the next Figure.

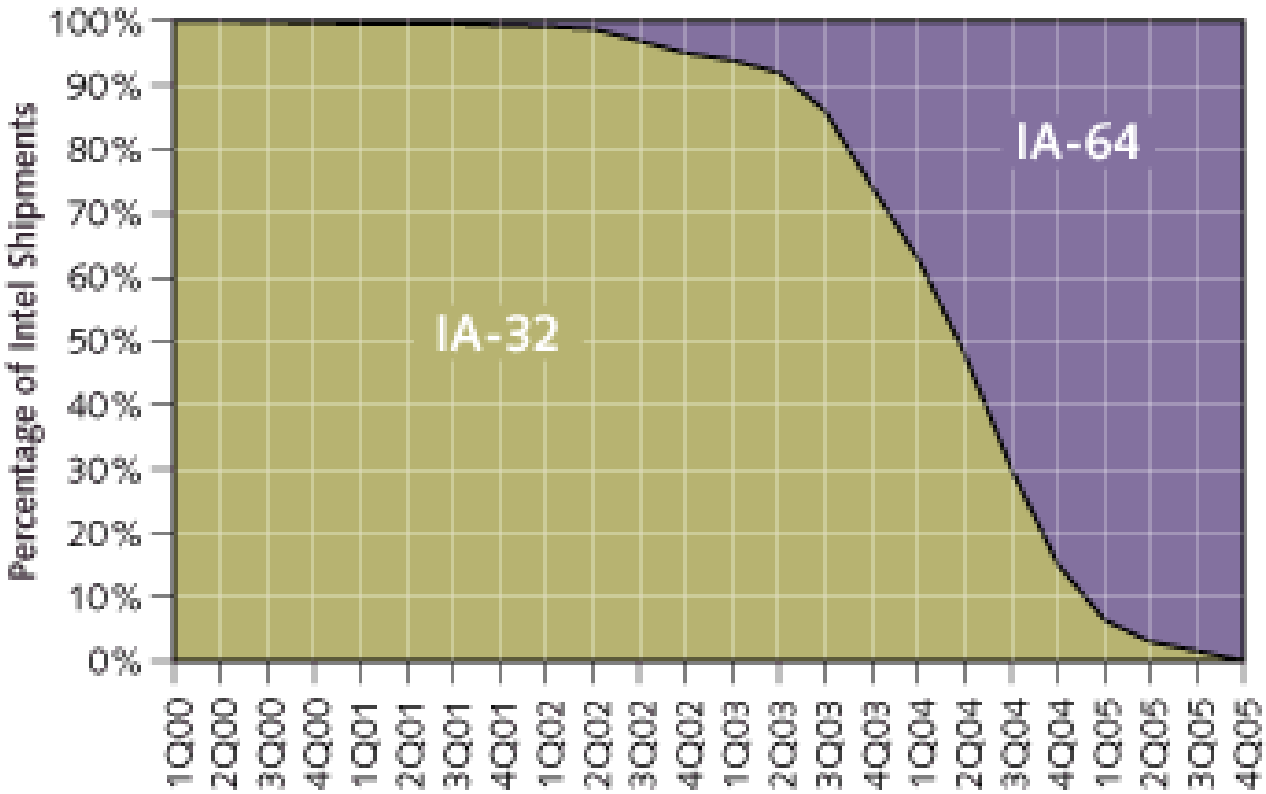


Figure: Prediction of an industry analyst for how IA-64 will supplant IA-32 [9]

Market reception of the first IA-64 processors-3

By contrast, due to the two years delay in shipping and the worse than expected performance sales expectations radically flattened in time, as the next Figure shows.

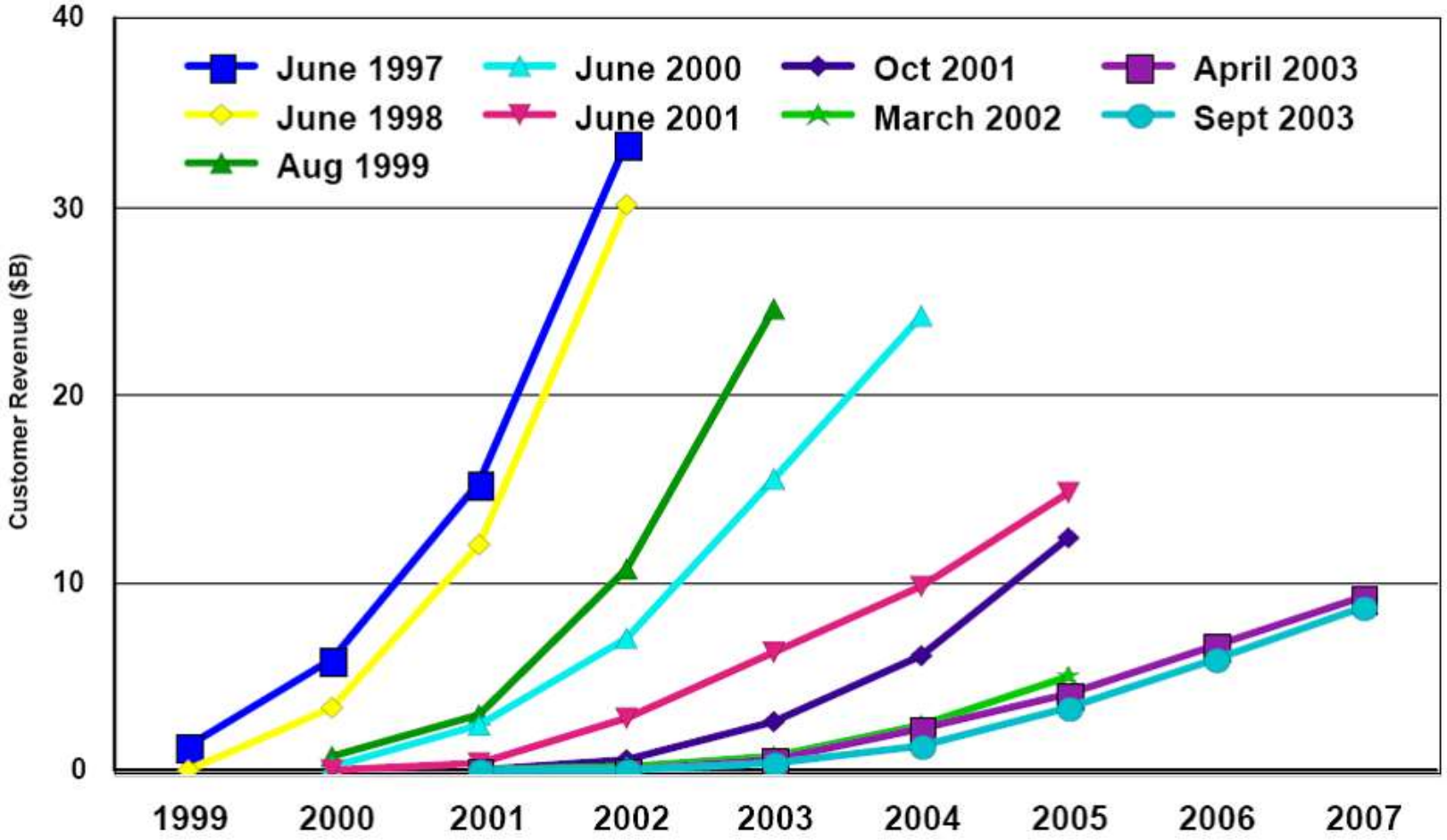


Figure: Sales expectations of the Itanium lines [10]

Market reception of the first IA-64 processors-4 [8]

Nevertheless, Compaq and Silicon Graphics decided to abandon further development of their Alpha and MIPS architectures, respectively in favor of migrating to IA-64.

A further fact contributing to abandoning RISC processors was that between 1995 and 2000 CISC processors overtook the performance leadership from CISCs, as indicated in the next Figure.

Shift of FX performance leadership from RISC processors to CISCs in 1995 - 2000 [68]

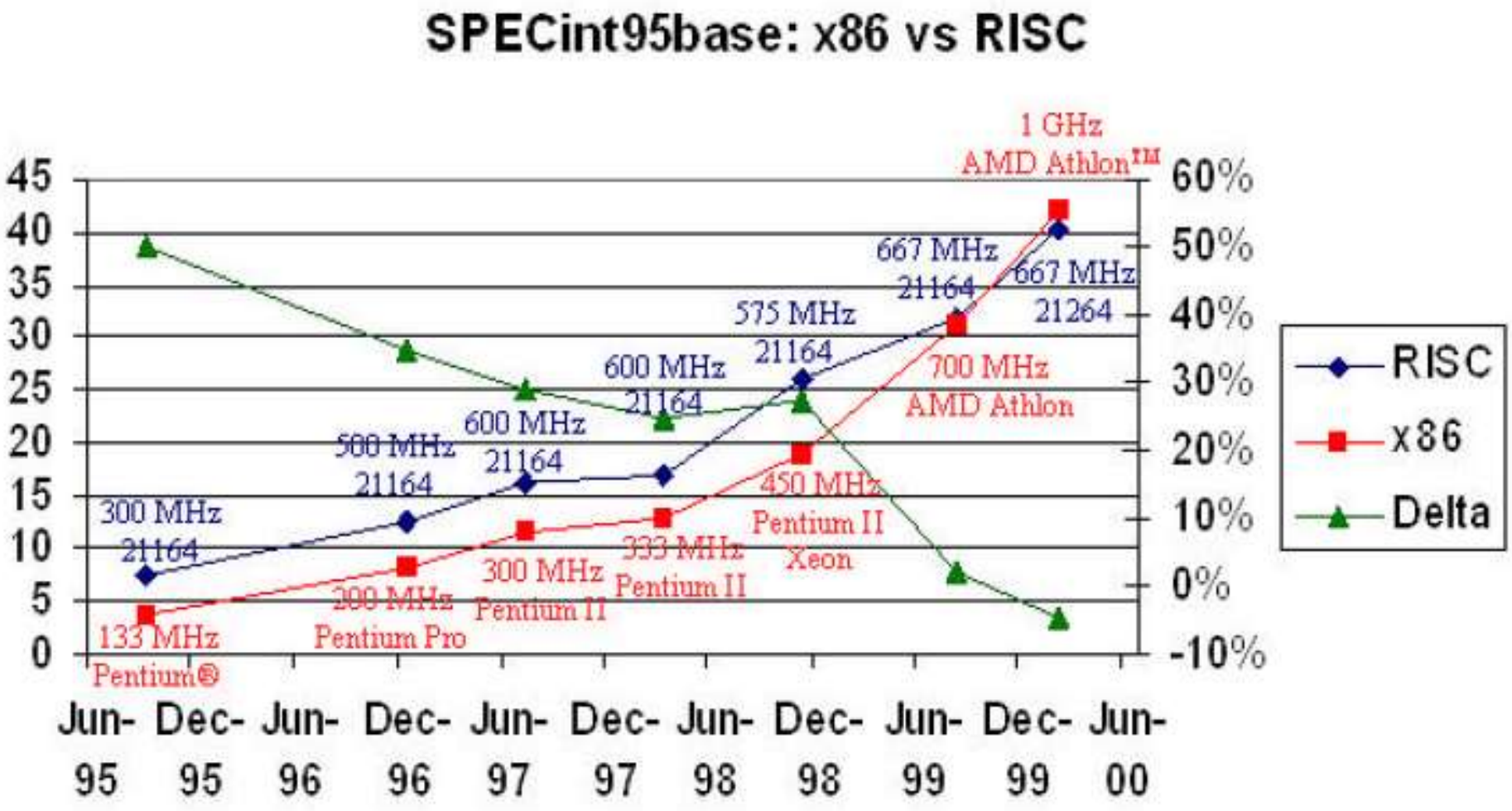


Figure: CISC processors supplanting RISCs between 1995 and 2000 [68]

2. The IA-64 ISA underlying the Itanium family

2. The IA-64 ISA underlying the Itanium family (1)

Remarks to the terminology

In these slides we will understand

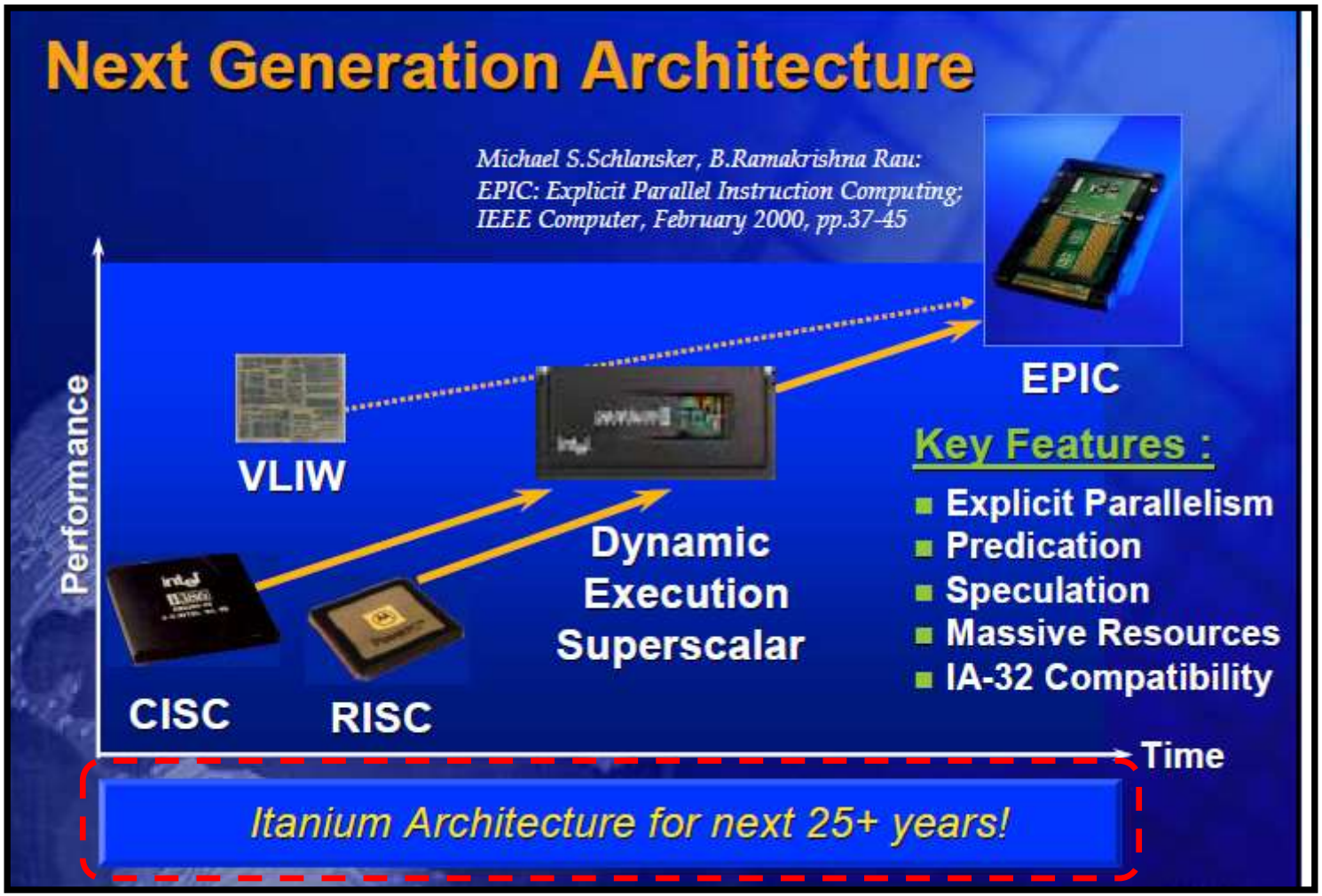
- the terms IA-64 as addressing the ISA of Itanium processors
- the term Itanium as covering the implemented family of processors and
- the acronym EPIC (Explicitly Parallel Instruction Computing) as a term addressing the principle of operation of both the ISA and the Itanium processors.

Nevertheless, in the literature the terms IA-64, EPIC and Itanium are often used as synonyms.

Note that we use the terms ISA and architecture as synonyms.

2. The IA-64 ISA underlying the Itanium family

Intel's view about the evolution path leading to EPIC architectures [11]



Motivation for developing EPIC architectures [12]

Many key designers of commercial VLIW machines became active in the EPIC project.

As stated in [12], they intended to develop a new style of processors while

- retaining the static instruction scheduling used by the VLIW processors to achieve high ILP with reduced hardware complexity since then no hardware dependency resolution is needed,
- use in-order instruction issue to minimize hardware complexity further more,
- to address general purpose computing as well thus being capable of achieving high levels of ILP on both numerical and scalar applications, in contrast to VLIW machines that targeted only numerical computing,
- augment the new design with advanced features introduced in the meantime into superscalar processors to better cope with dynamic factors of the execution and
- provide adequate x86 object code compatibility across evolving families of processors, as required for general-purpose processors.

The EPIC (Explicitly Parallel Instruction Computer) principle of operation

It covers three main design decisions as follows:

a) Static (compiler performed) instruction scheduling
b) exposing parallelism explicitly (through Stop bits) and
c) in-order issue of instructions from two bundles to the available issue ports,

as discussed next.

2. The IA-64 ISA underlying the Itanium family (5)

a) Static (compiler performed) instruction scheduling-1

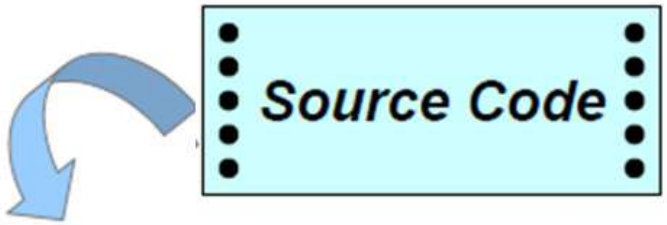
In order to relieve hardware from carrying out dependency checking and thus to simplify it, the compiler is expected to

- analyze code for dependencies among instructions,
- optimize code for parallel execution and
- schedule instructions for execution into instruction groups, called bundles such that
  - each bundle includes three instructions,
  - all three instructions in a bundle may be be executed in parallel and
  - each bundle is attached a tag, called the template to indicate hardware requests and expose dependencies between bundles,

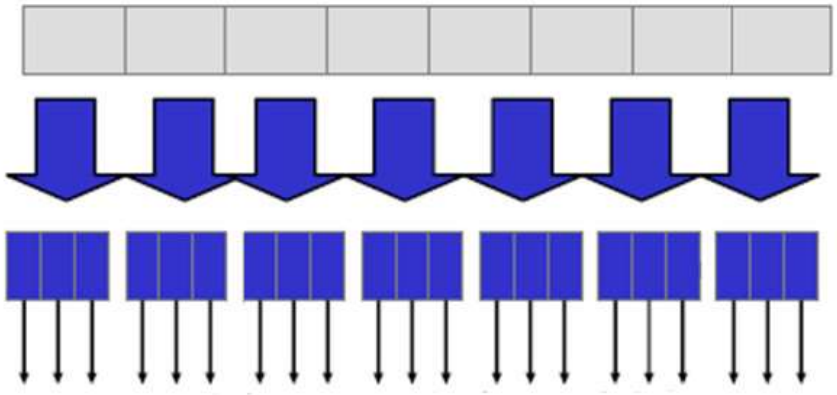
as indicated in the next Figure and described subsequently.

a) Static (compiler performed) instruction scheduling-2 (based on [13])

Compiler



Instruction bundles



Three instructions each

There are no dependencies between the instructions within a bundle

b) Exposing parallelism explicitly (through Stop bits)

The compiler makes use of so called Stop bits to expose dependencies between subsequent bundles.

The Stop bits are kept in the Template part of the bundles, and identify the set of bundles that can be executed in parallel, as shown below.

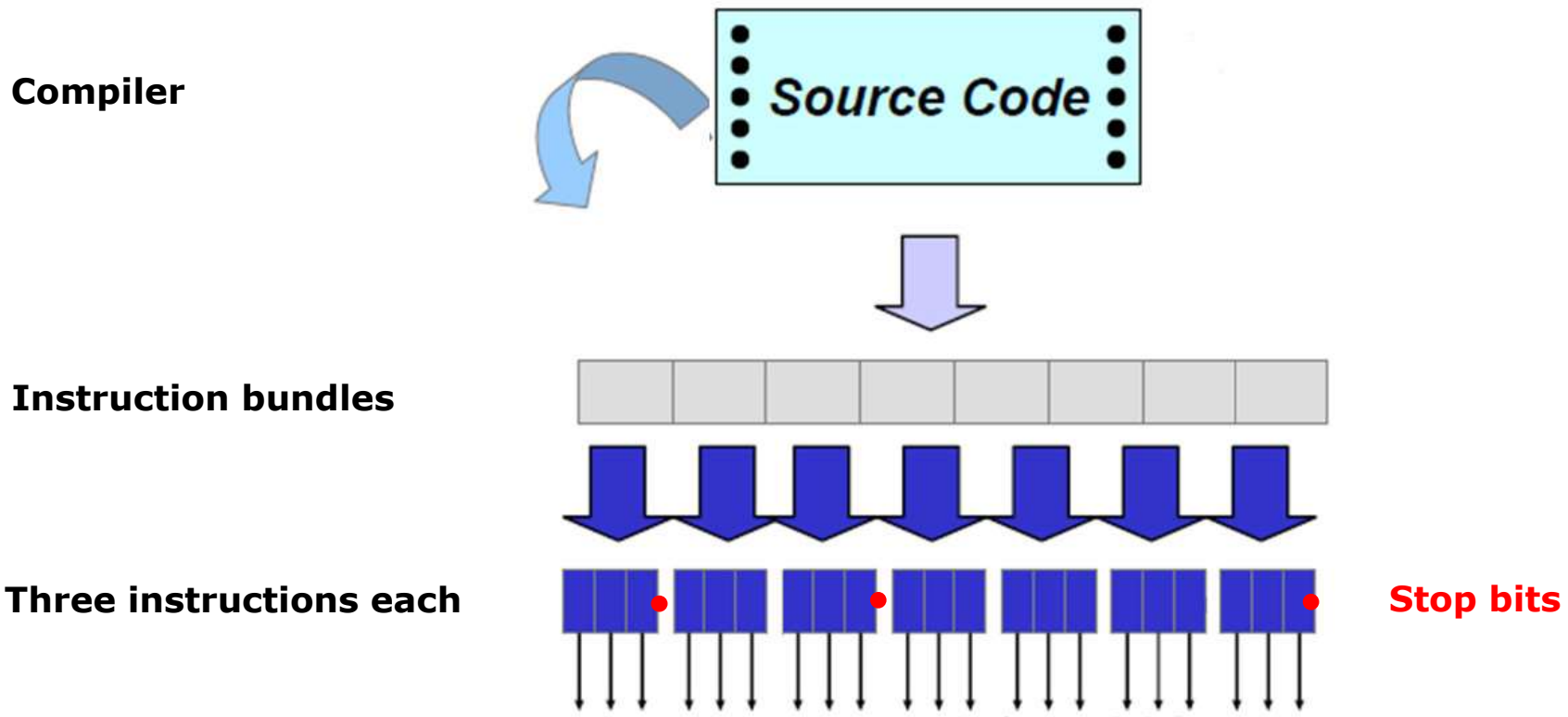


Figure: Exposing parallelism explicitly through Stop bits (based on [13])

c) In-order issue of instructions from two bundles to the available issue ports

Instructions are issued from two bundles in-order until

- a Stop bit indicates dependency with the next bundle or
- a requested input port is not available.

After all three instructions of a bundle have been issued a new bundle is loaded from the instruction buffer.

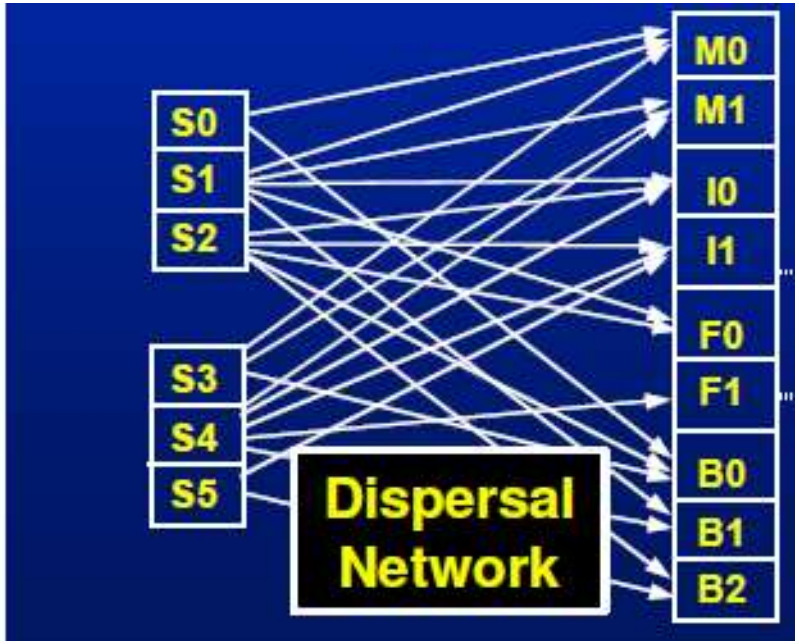
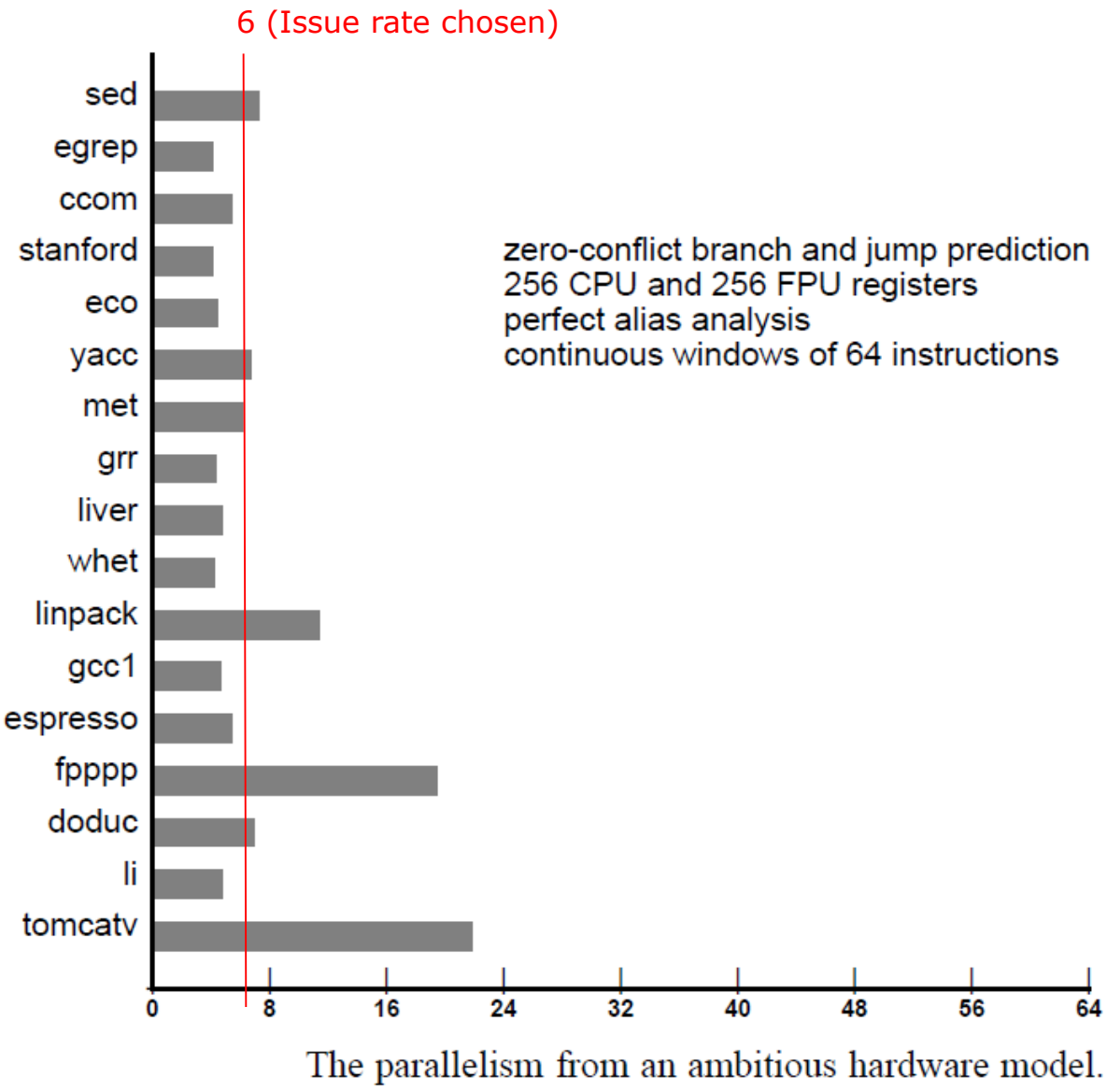


Figure: instruction issue in case of the first Itanium processor [14]

Remark to the design decision to issue instructions from two bundles-1

- There is a seminal research paper of Wall concerning the limits of available parallelism in applications [15].
- In the next Figure we recall the main results of his investigation and contrast it with the design decision to issue up to 6 instructions.

Limits of ILP according to Wall's early investigations [15]



The parallelism from an ambitious hardware model.

Remark to the design decision to issue instructions from two bundles-2

According to Wall's cited results and taken for granted that IA-64 addresses also general purpose computation the design decision to issue instructions from two bundles appears to be optimum, as taking only one bundle would constrain the issue potential for most applications whereas taking three bundles would waste hardware resources for general purpose computations.

Key features of the IA-64 ISA and implemented microarchitectures [11]

a) 64-bit ISA and microarchitecture with a wide range of HW supported data types targeting both 64-bit general purpose and numerical computing

b) large register space to support the advanced techniques introduced to raise ILP (see Section d)

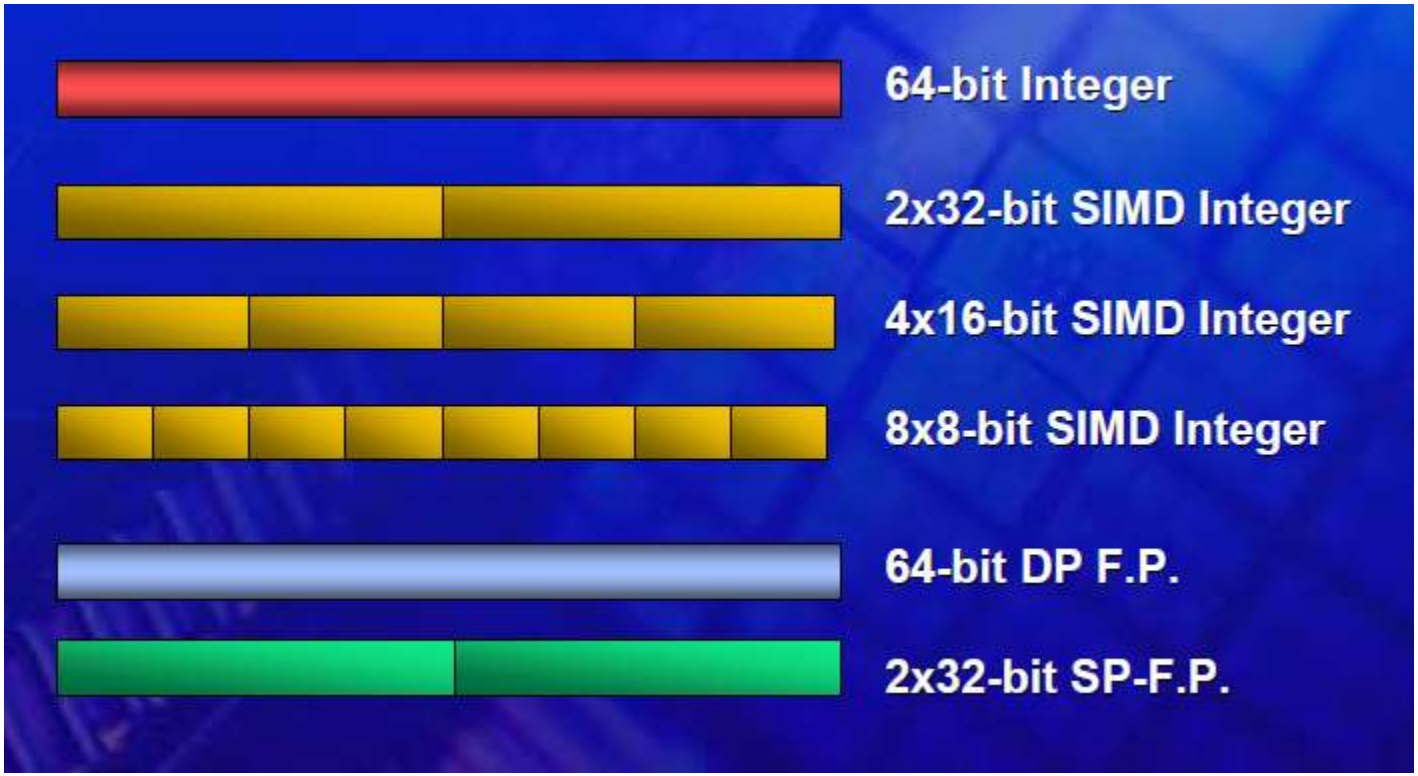
c) RISC-type (i.e. load-store) instruction semantics

d) supporting a number of advanced techniques to increase ILP both in the ISA and the microarchitecture

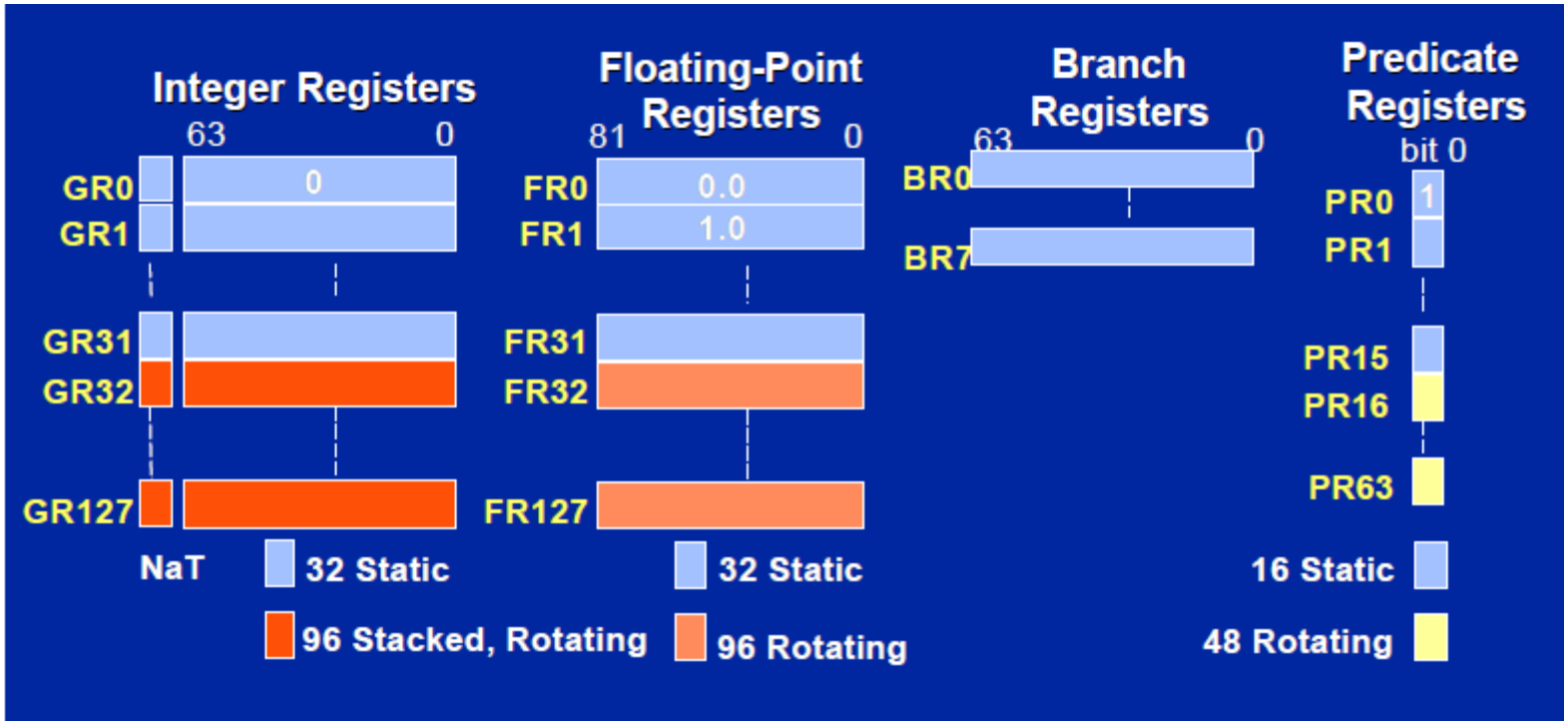
e) providing IA-32 compatibility.

2. The IA-64 ISA underlying the Itanium family (13)

a) 64-bit ISA and microarchitecture with a wide range of HW supported data types targeting both 64-bit general purpose and numerical computing [1



b) Large register space to support the advanced techniques introduced to raise ILP (see Section d) [17]



c) RISC-type (i.e. load-store) instruction semantics [18]

Instructions are of RRR type with providing 7-bits for addressing a register space of 128 registers.

The next Figure shows the chosen instruction and bundle formats.

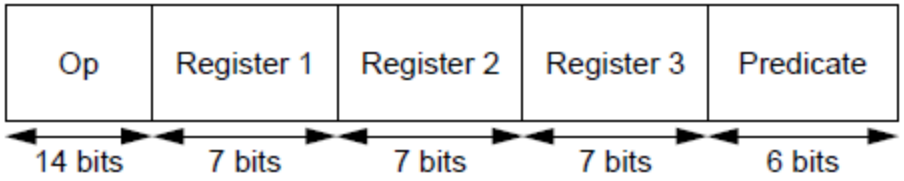


Figure: IA-64 instruction format [18]

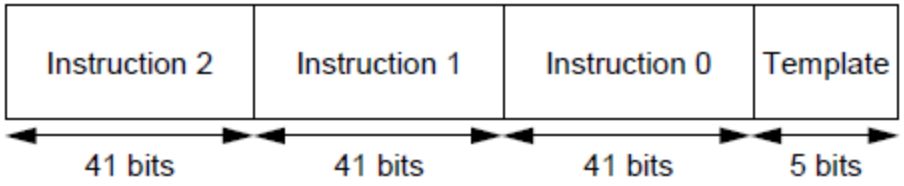


Figure: IA-64 bundle format [18]

The template bits help to decode and route instructions to the issue pipelines and indicate the location of stops that mark the end of bundles that can be executed in parallel.

d) Supporting a number of advanced techniques to increase ILP both in the ISA and the microarchitecture

The IA-64 ISA and related processor microarchitectures support a number of advanced techniques to increase ILP.

From these techniques subsequently we will discuss briefly only the following ones:

- d1) Predication
- d2) Speculation
- d3) Support of register stacks
- d4) Support of software pipelining
- d5) Hints for branches and caching.

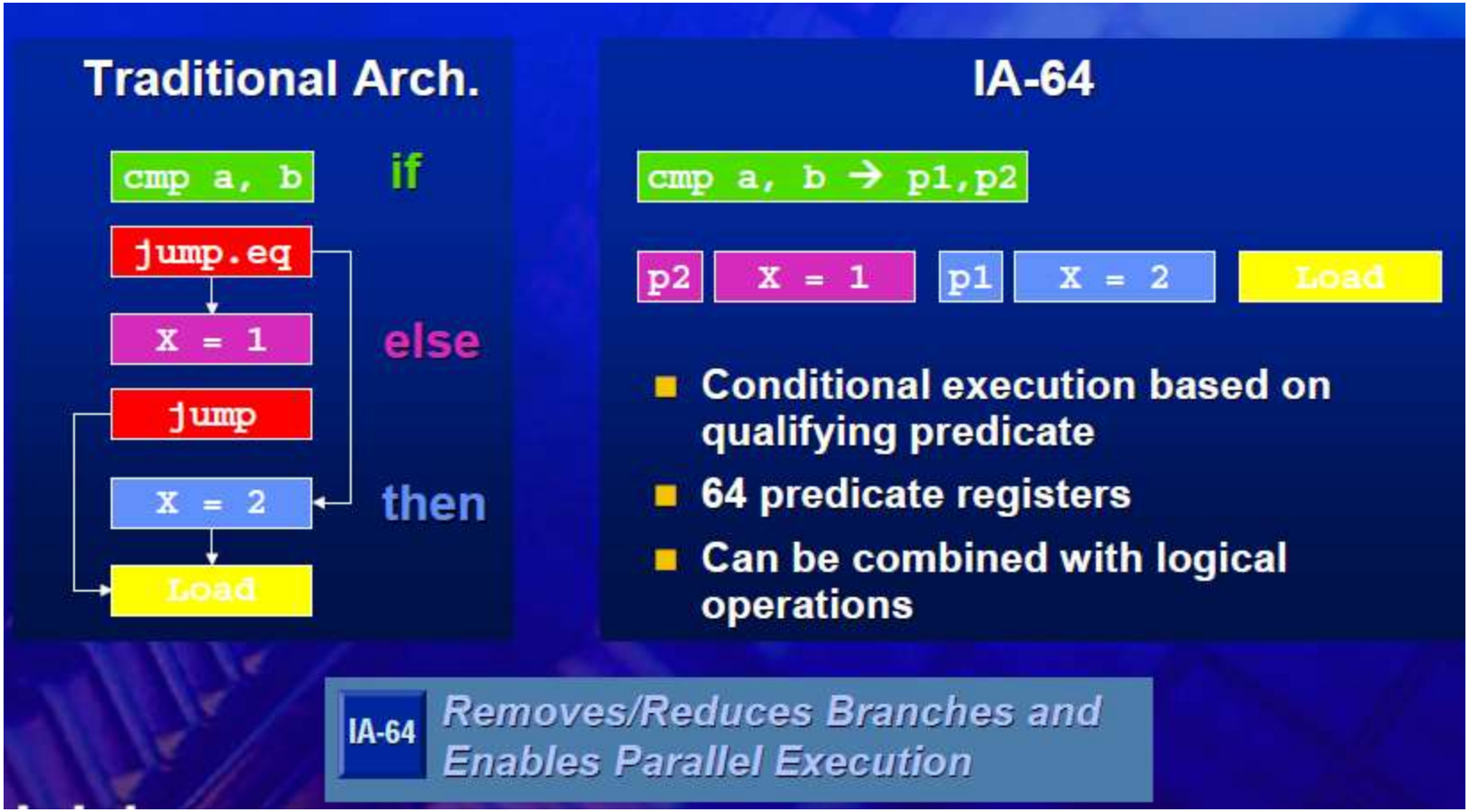
For a comprehensive discussion of the set of introduced techniques we refer to [12], [18].

d1) Predication [12], [18]

- Predication is introduced to remove branches and enable parallel execution.
- Prediction makes use of 64 1-bit predicate registers.
- With predication conditional branches are substituted by compare operations such that the result of the evaluation of the conditional expression ("true" or "false") will be saved in selected 1-bit predicate registers.
- Instructions are then guarded (predicated) meaning that instruction execution depends on the value of the referenced predicate register such that an instruction will be executed only if the referenced predicate value is "true".
- Seeing from another point of view predication substitutes control flow by data flow.
- In addition, with predication both sides of a conditional branch can be executed in parallel when there are sufficient processor resources available.

The next Figure illustrates this.

Example for predication [19]



The cmp (compare) instruction generates two predicates (p1) and (p2) that are set to one or zero, based on the result of the comparison (p1 will be set to the opposite of p2).
These predicates can be used to guard execution: the x = 1 instruction will only execute if p2 has a true value, and the x = 2 instruction will only be executed if p1 has a true value [18].

Remark

In the above example the result of the comparison is forwarded to two predicate registers (p1) and (p2).

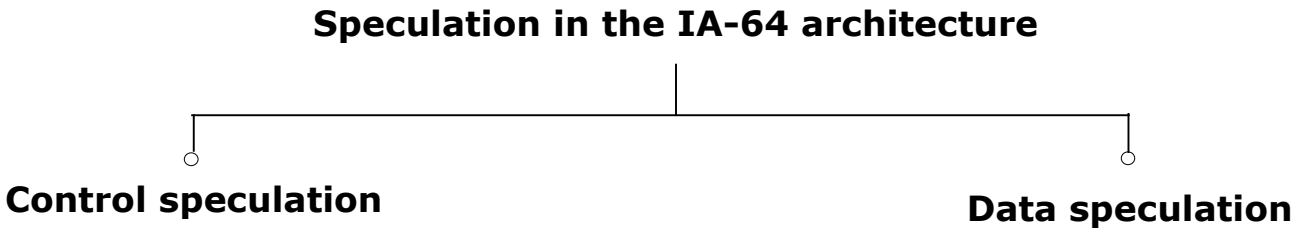
If the instruction format would allow negations (i.e. e.g. p1 could be specified) less predicate register would be needed but hardware complexity of the control circuitry would be raised.

Presumably, all in all the chosen solution is more optimal.

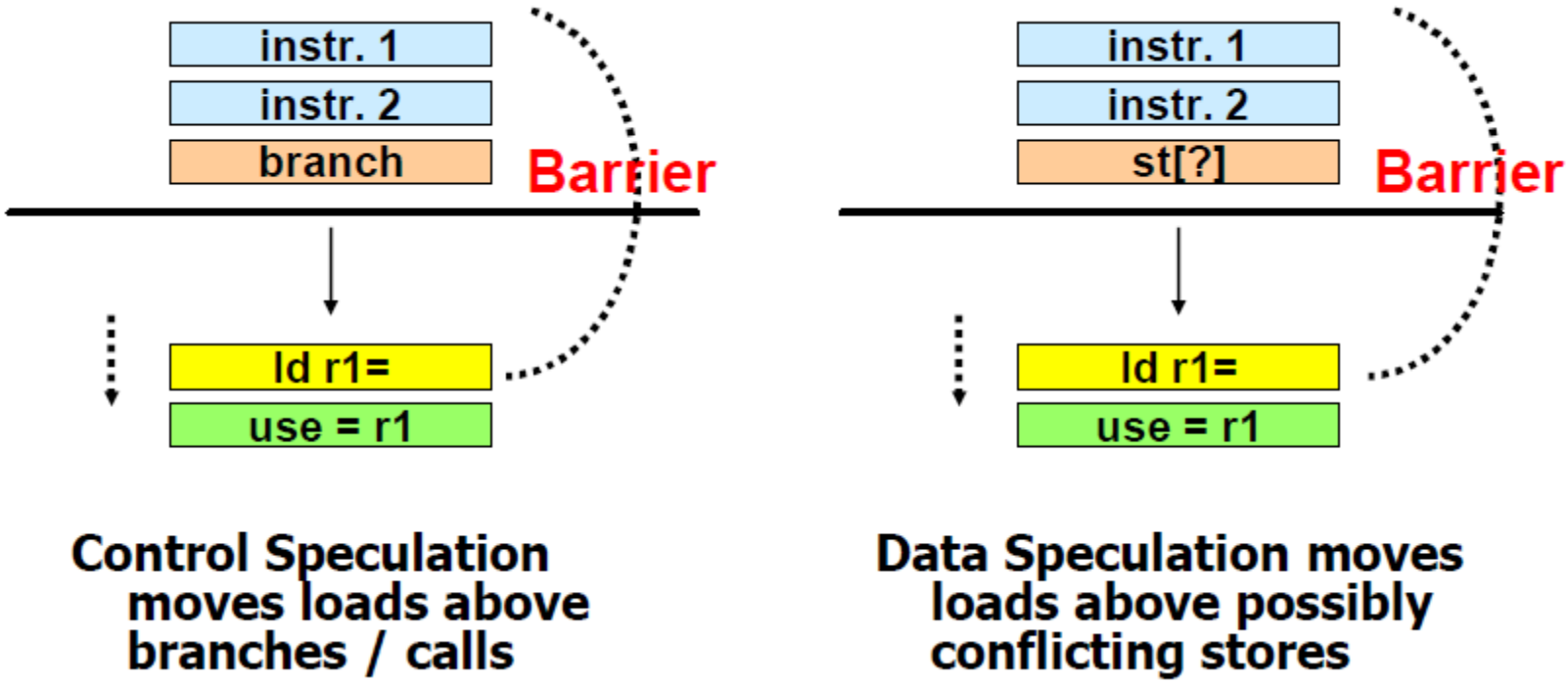
d2) Speculation

Speculation enhances the compiler to generate more efficient EPIC schedules by allowing various forms of aggressive, compile-time code motion that would be illegal in a conventional architecture [12]

There are two types of speculation; control speculation and data speculation, as indicated below and will be discussed subsequently.



Overview of the principles of control and data speculation [13]



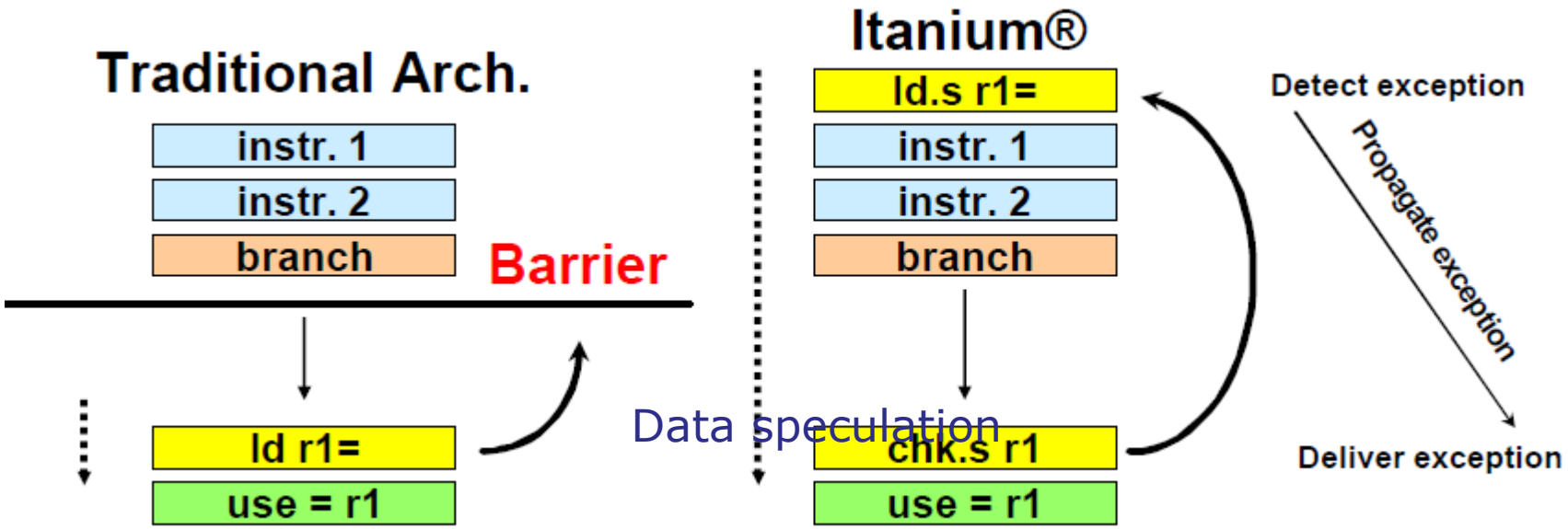
Speculation reduces the impact of memory latency

Control speculation [13]

- IA-64 provides a new group of loads, called speculative loads (ld.s) which can safely be scheduled before one or more prior branches.
- In the block where the programmer originally placed the load, the compiler schedules a speculation check (chk.s), as shown in the next Figure.
- At runtime, if a speculative load results in an exception, the exception is deferred and a token (NaT) is written to the target register.
- The chk.s instruction checks then the target register for a NaT token, and if a token is present the execution branches to a special "fix-up" code.
- If needed the "fix-up" code will re-execute the load non-speculatively.

The next Figure gives an example for this.

Example for control speculation [13]



- **Control Speculation moves loads above branches**
  - **Detected exception indicated using NaT bit / NaTVal**
- **Check raises detected exceptions**
- **Branch barrier broken to minimize memory latency**

NaT and NaTVal are considered to be tokens rather than data values used in FX and FP data, respectively.

Data speculation [12]

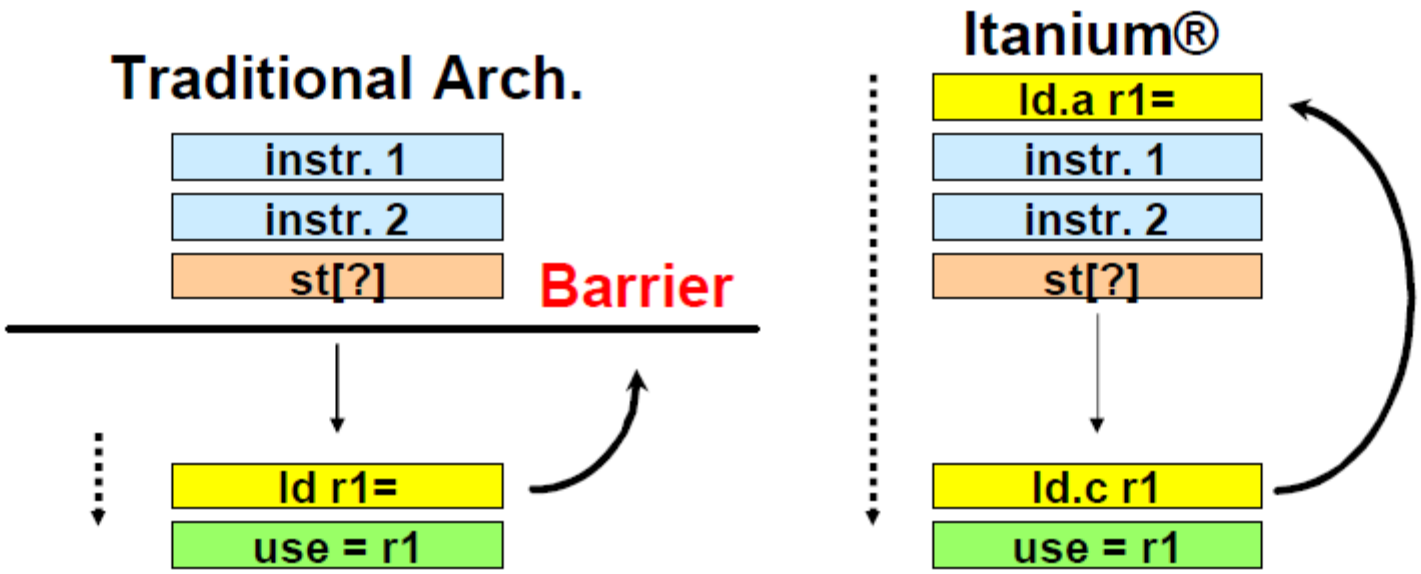
- Data speculation substitutes a conventional load by a advanced load (ld.a) and a data- verifying load (ld.c).
- The compiler can move a data-speculative load above potentially aliasing stores to allow the load an early schedule.
- The compiler schedules the data verifying load in place of the conventional load.
- The processor uses hardware to detect whether an alias has occurred.

  When no alias has occurred, the data-speculative load has already loaded the correct value.

  In the opposite case, the data-verifying load re-executes the load and stalls the processor to ensure that the correct data returns in time for subsequent uses of the load.

The next Figure illustrates data speculation.

Example for data speculation [13]



- **Data Speculation moves loads above possibly conflicting stores**
  - **Keeps track of load addresses used in advance (ALAT)**
- **Advanced-loaded data can be used speculatively**

- ALAT is the Advanced Load Address Table in Itanium architectures.
- It is implemented with an associative memory and is used to store information related to advance load instructions, to implement data speculation [20].

d3) Support for register stacks [21], [22]

Aim and introduction of register stacks

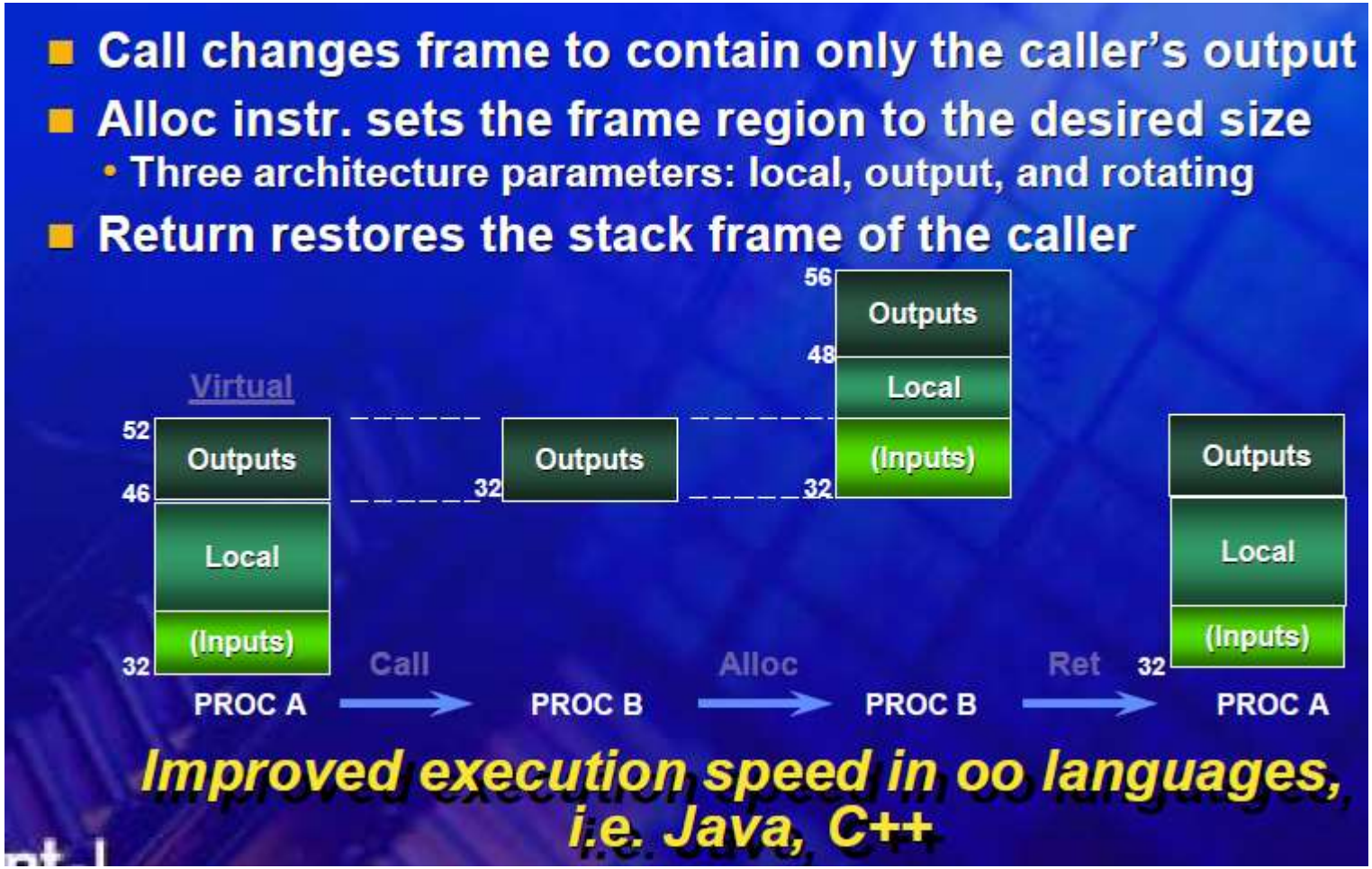
- Register stacks aim at speeding up argument passing in procedure calls.
- This technique was introduced in the original Berkeley RISC design (~ 1980), termed as register windows, and used subsequently in a number of further designs, such a SPARC (1986) or AMD 29000 (1988) with different features.

Principle of operation of register stacks

- During procedure calls each procedure gets allocated a set of registers, subdivided into input, local and output registers.
- Nevertheless, for embedded procedure calls the input registers are aliased with the output registers of the calling procedure, thus no operand transfer is needed between procedures.
- Returns restore the register stack of the caller.

The next Figure illustrates the operation of register stacks.

Example for register stacks [16]



d4) Support for software pipelining [18]

Aim and principle of supporting software pipelining

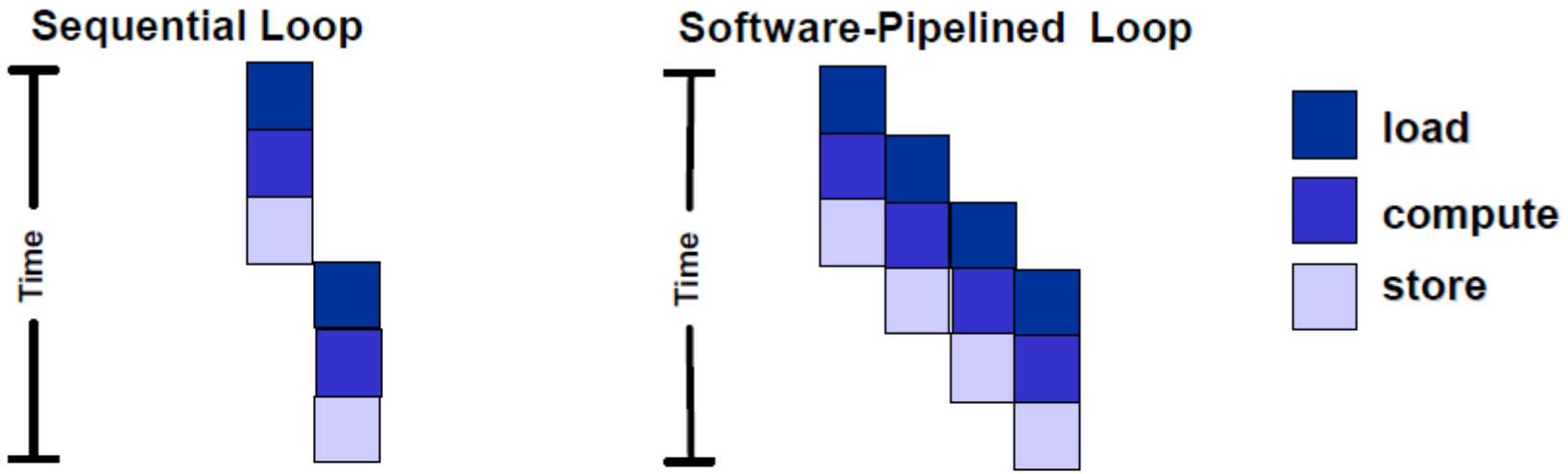
- In numeric computations software pipelining (loop unrolling) is often used to speed up the execution of loops, for details see e.g. [23].

  Nevertheless, this often causes a significant code size expansion.
- IA-64 eliminates most of the overhead associated with software pipelining.
- To achieve this IA-64
  - provides special registers to keep the loop count (LC) and the pipeline length (epilog count or (EC)),
  - allows to use a subset of the general, FP- and predicate registers (called rotating registers) to be automatically renamed after each iteration by decrementing a register rename base (rrb) register.
  - Then for each rotation, all the rotating registers appear to move up one higher register position, with the last rotating register wrapping around the bottom.

    Thus each rotation effectively advances the software pipeline by one stage.

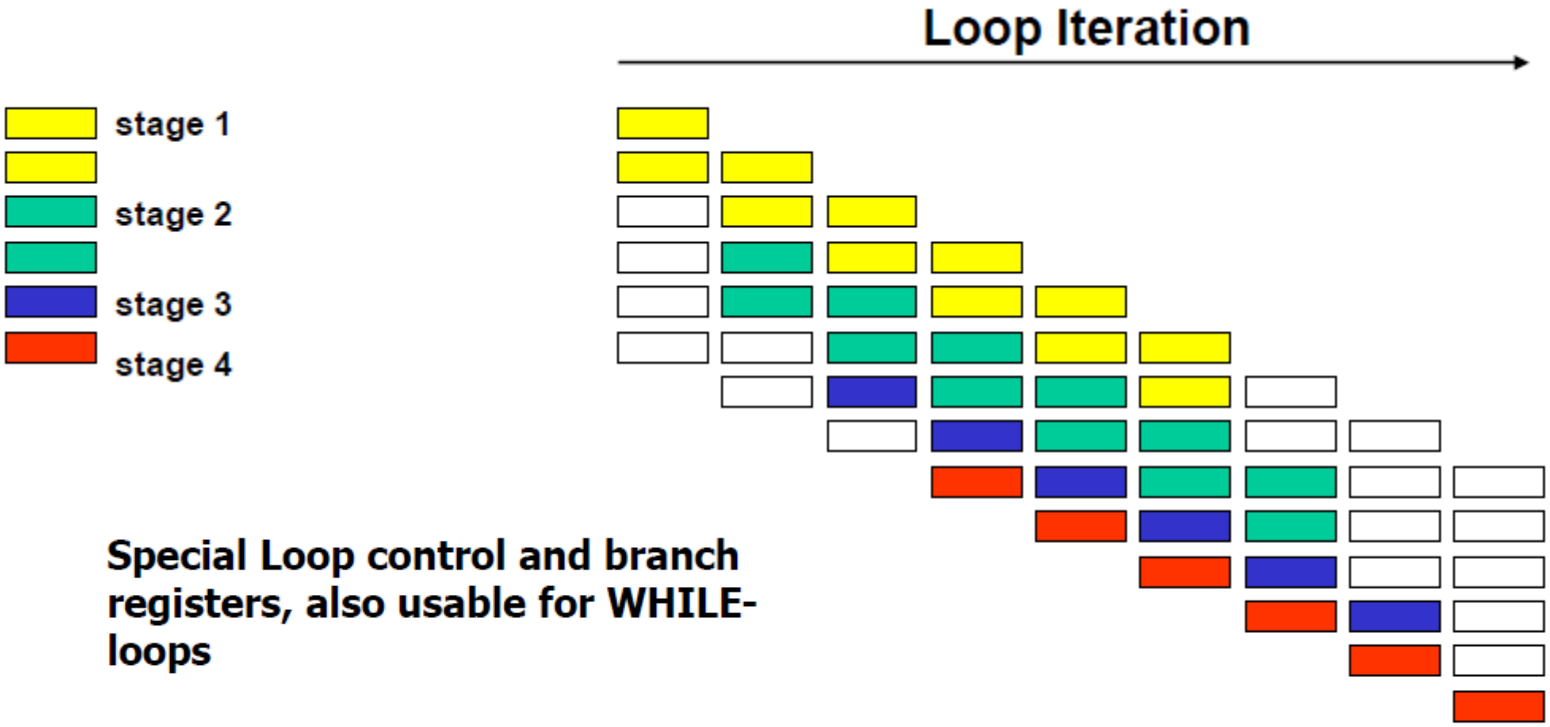
The next Figures illustrate this.

Support for software pipelining-2 [13]



- Traditional architectures use loop unrolling
  - Results in code expansion and increased cache misses
- Itanium™-Processor Software Pipelining uses rotating registers
  - Allows overlapping execution of multiple loop instances
- Predication controls the pipeline stages

Support for software pipelining-3 [13]

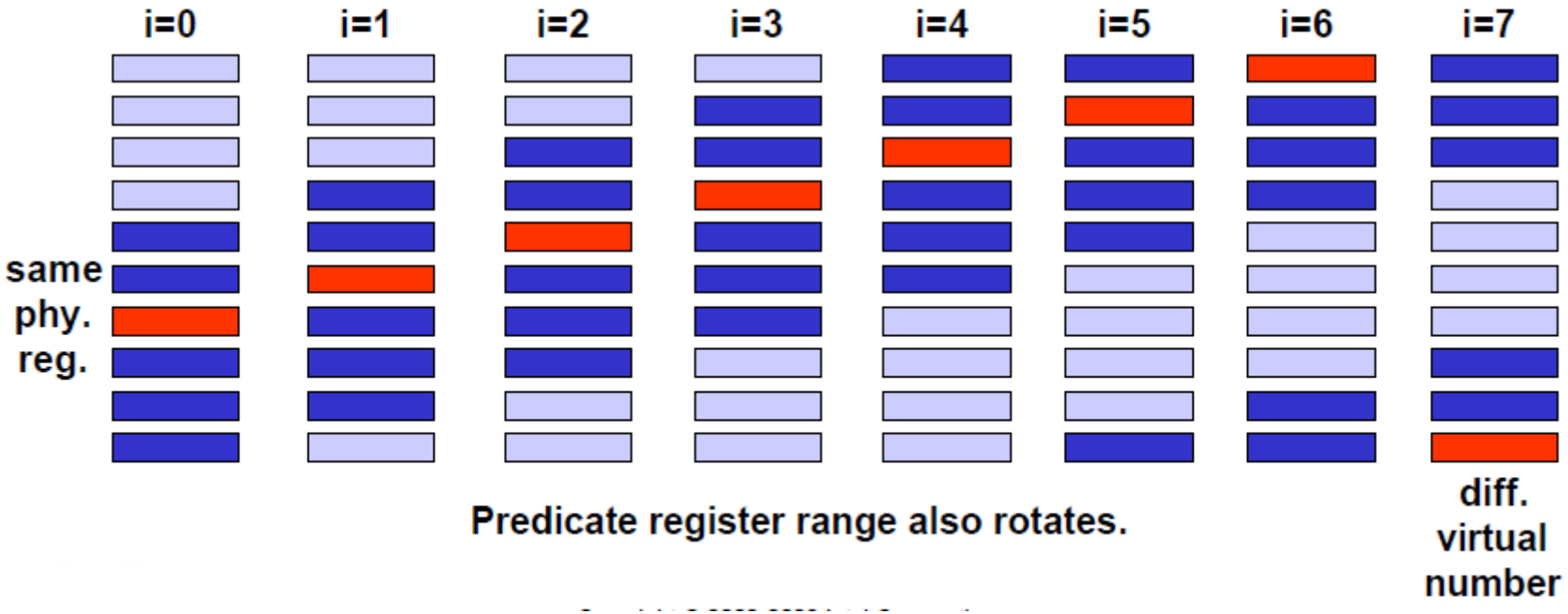


Special Loop control and branch registers, also usable for WHILE-loops

Predicate registers rotate as well and define the pipeline stages

Register rotation [13]

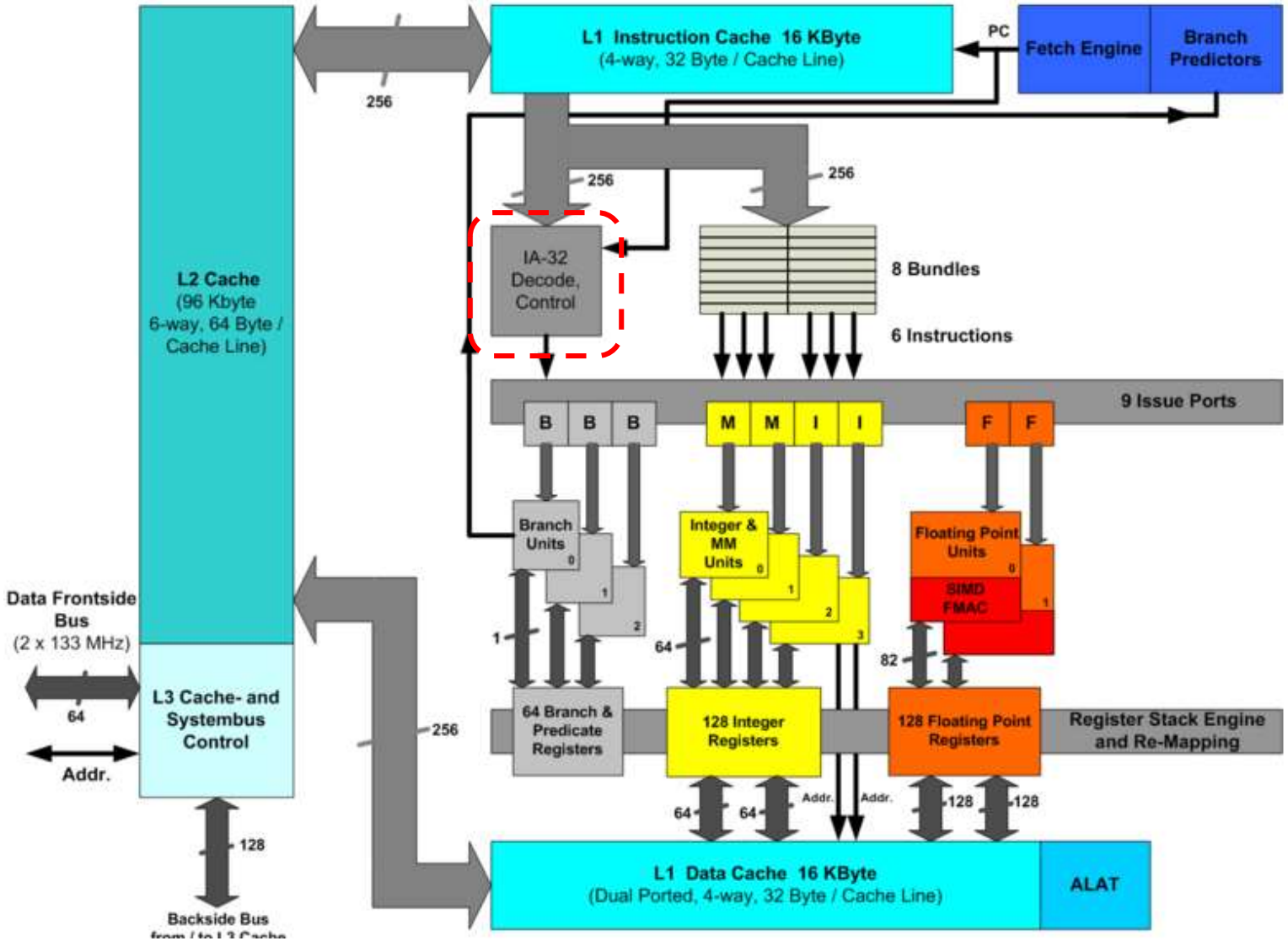
- GR32-127 and FR32-127 can rotate (specified range)
- Separate rotating register base for each set (GR, FR)
- Loop branches decrement all register rotating bases (RRB)
- Instructions contain a "virtual" register number
  - physical register # = RRB + virtual register #



e) Providing IA-32 compatibility-1 [24]

- First IA64 processors (including Itanium, Itanium McKinley and Madison 6 MB) provided x86 compatibility by a dedicated hardware unit, as indicated in the next Figure for the first Itanium processor (Merced).

Execution of x86 code on the Itanium (Merced) core-1 [25]

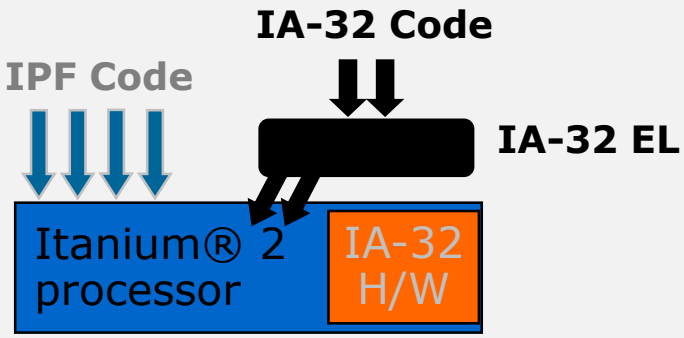


Execution of x86 code on the Itanium (Merced) core-2 [24]

- IA-32 compatibility includes support for running a mix of IA-32 and IA-64 applications on an IA-64 OS, as well as IA-32 applications on an IA-32 OS.
- The IA-32 engine makes use of Merced's registers, caches and execution resources.
- First Itanium processors executed IA-32 (x86) code extremely slow, according to reports x86 code run on a 667 MHz Itanium processor not faster than on a 75-100 MHz Pentium [26].

Introducing software emulation of x86 code in the Itanium 2 series-1 [27]

- In 1/2003 Intel introduced software emulation (dynamic translation) for the Itanium 2 series IA-64 processors by providing a software execution layer, called the IA-32 Execution Layer (IA-32 EL), as indicated in the Figure below.



- Historically, support of IA-32 applications has been carried out by on-die hardware
- When using operating systems with IA-32 EL, support for IA-32 applications will be provided by IA-32 EL
- IA-32 EL will ship with leading OSs
- Available 1/13/2003 with Windows Server 2003, Windows XP Professional

Figure: Principle and availability of the IA-32 Execution Layer (IA-32 EL) [28]

- As an application-level binary translator IA-32 EL runs above the 64-bit OS in the application program's virtual space.

  Once loaded and initialized, IA-32 EL gets control from the OS in order to run the 32-bit code of the application within the same virtual address space.

Introducing software emulation of x86 code in the Itanium 2 series-2 [27]

As software emulation proved faster than hardware supported x86 execution

Intel removed the hardware support for executing IA-32 code beginning with their Itanium 2 Madison 9 MB processor (2004) and provided x86 compatibility subsequently only by software emulation.

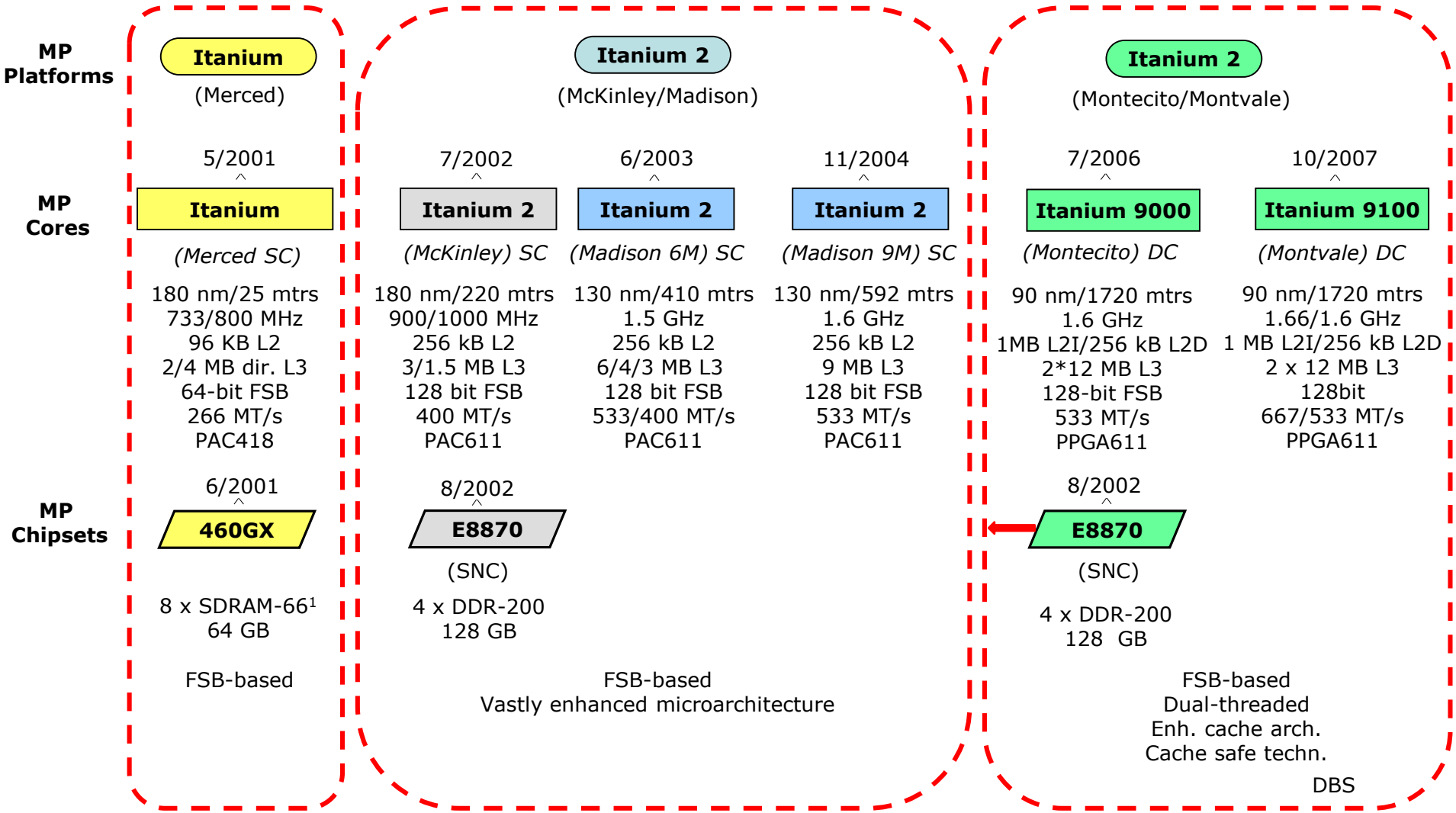
3. Overview of the evolution of the Itanium family

Early Itanium roadmap from 1999 [17]

Note: The Itanium processor (Merced) was scheduled for 2000 but appeared in 5/2001

3. Overview of the evolution of the Itanium family (2)

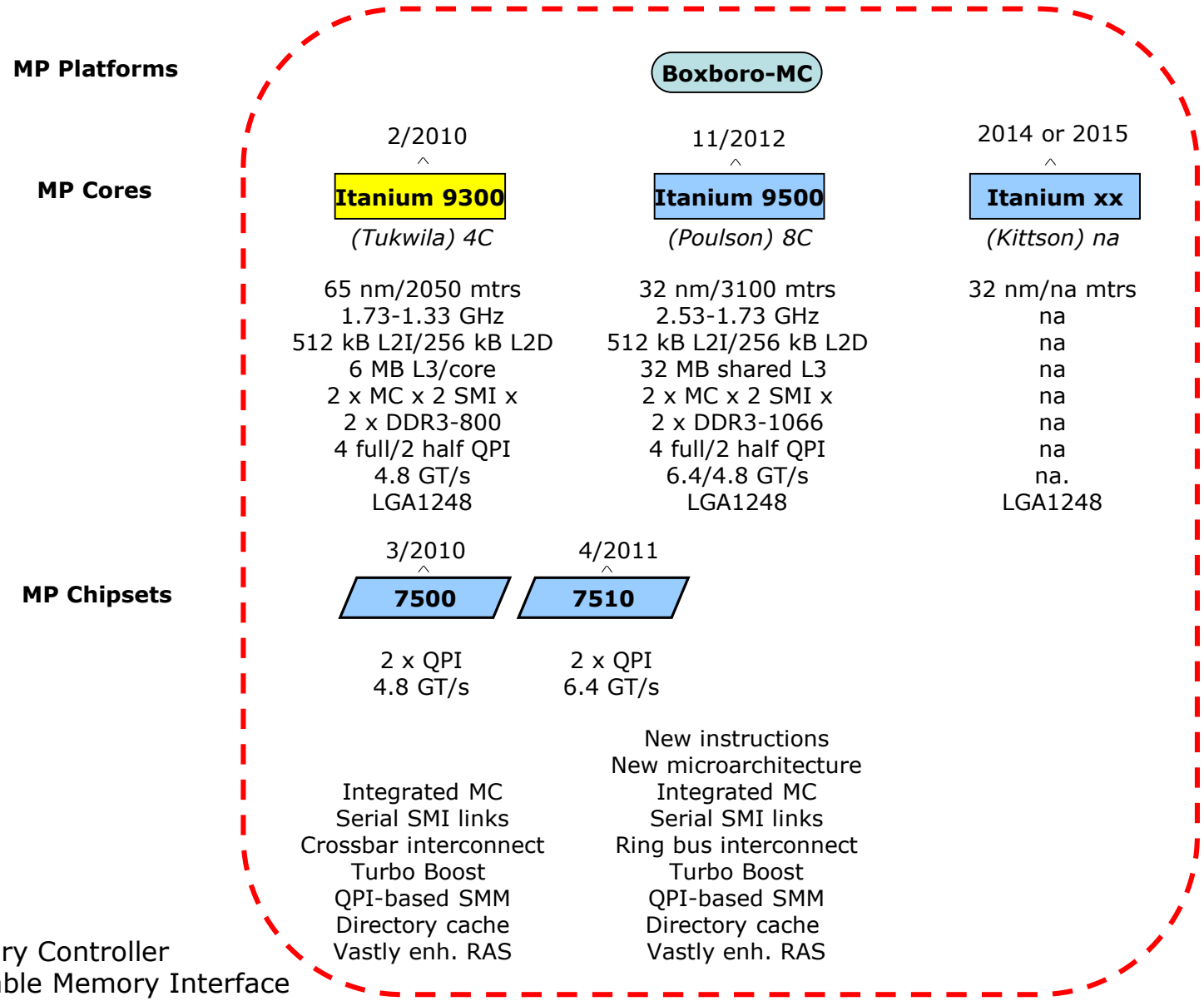
Overview of the Itanium and Itanium 2 platforms



[1]: Special memory cards are used SNC: Scalable Node Controller

3. Overview of the evolution of the Itanium family (3)

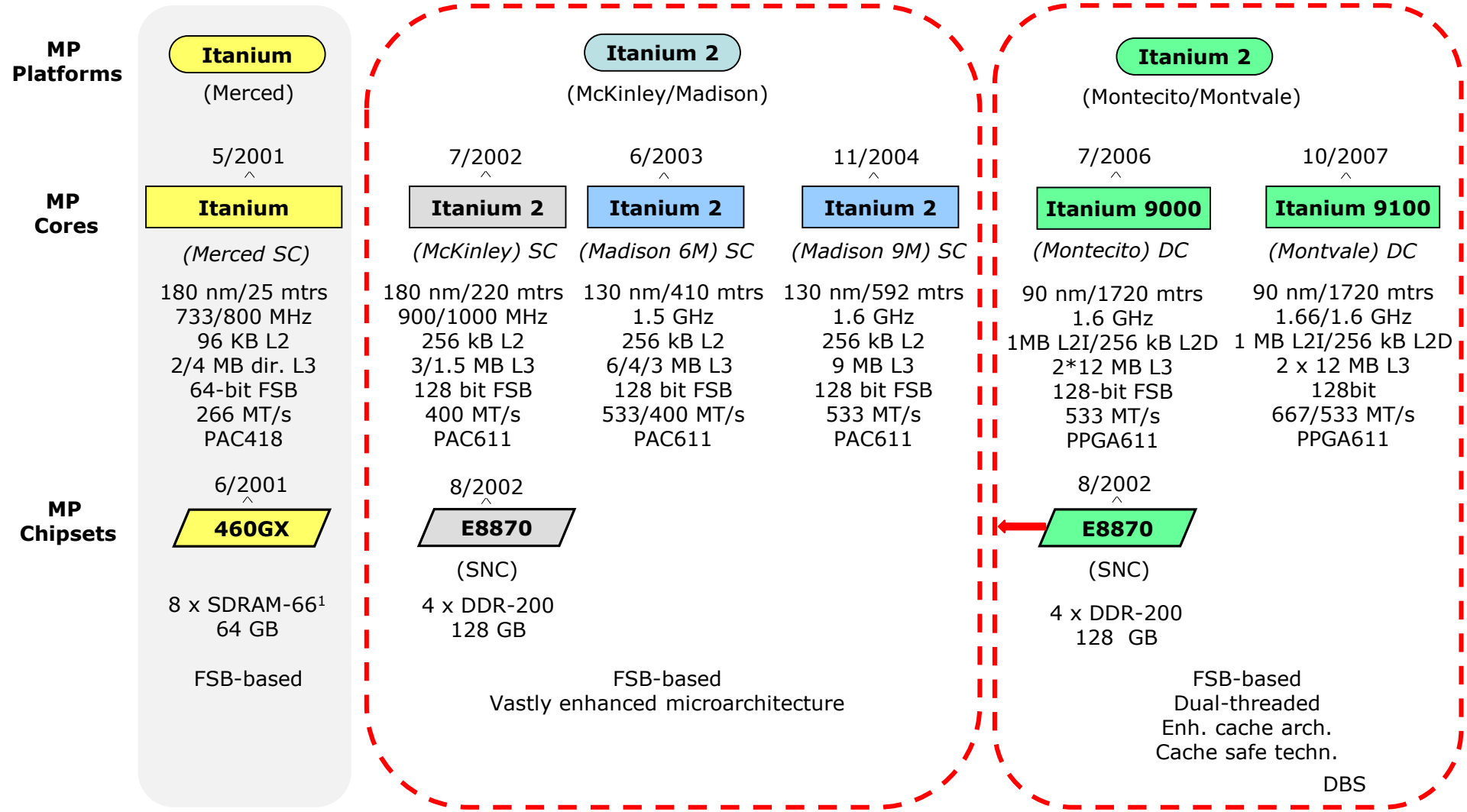
The Boxboro-MC platform



MC: Memory Controller
SMI: Scalable Memory Interface

4. The Itanium line

4. The Itanium line



[1]: Special memory cards are used

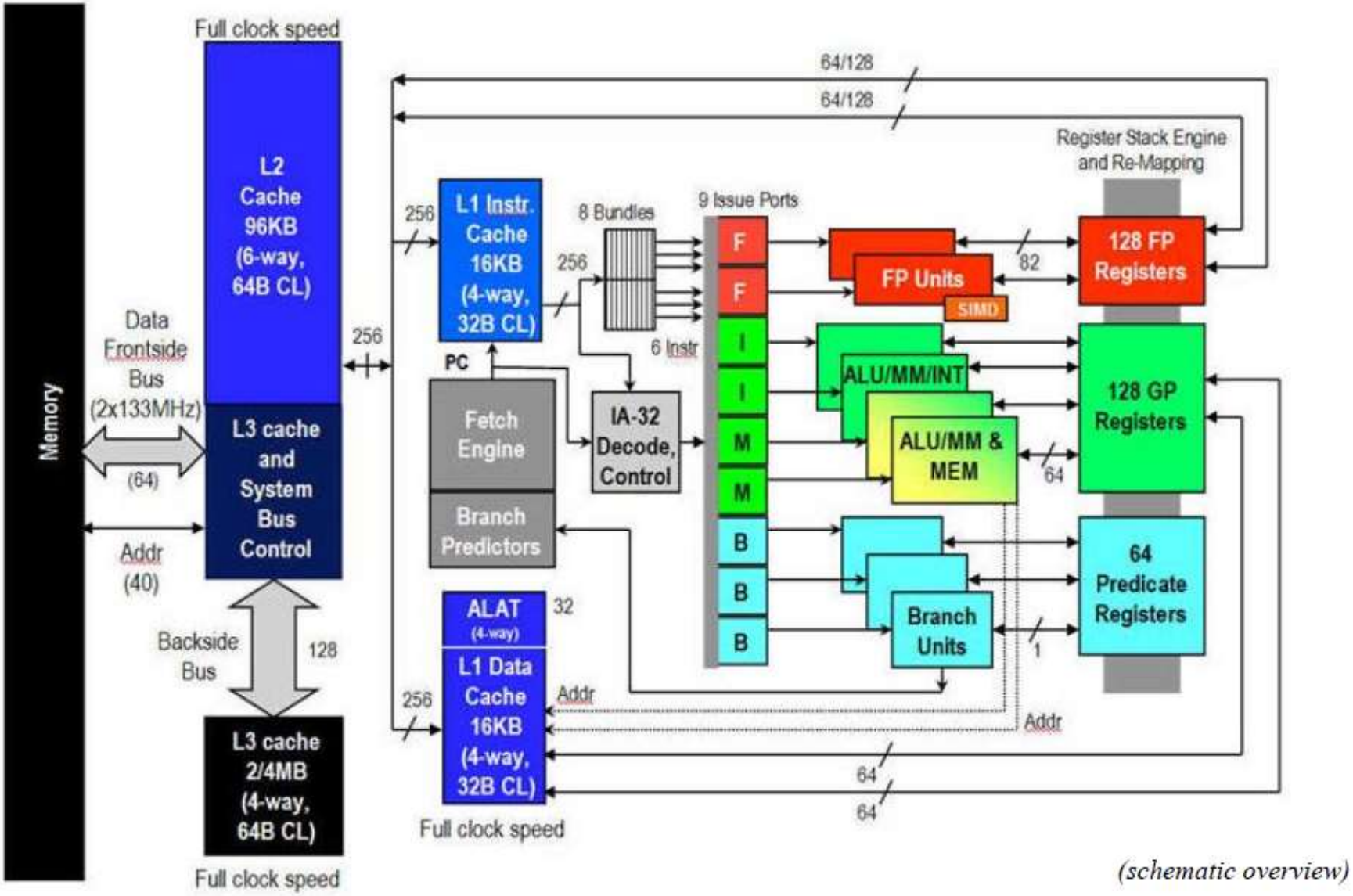
SNC: Scalable Node Controller

4. The Itanium line (2)

The Itanium line

- Originally scheduled for 1999 but launched only in 5/2001.
- Manufactured by 180 nm feature size.

Block diagram of the Itanium processor [13]

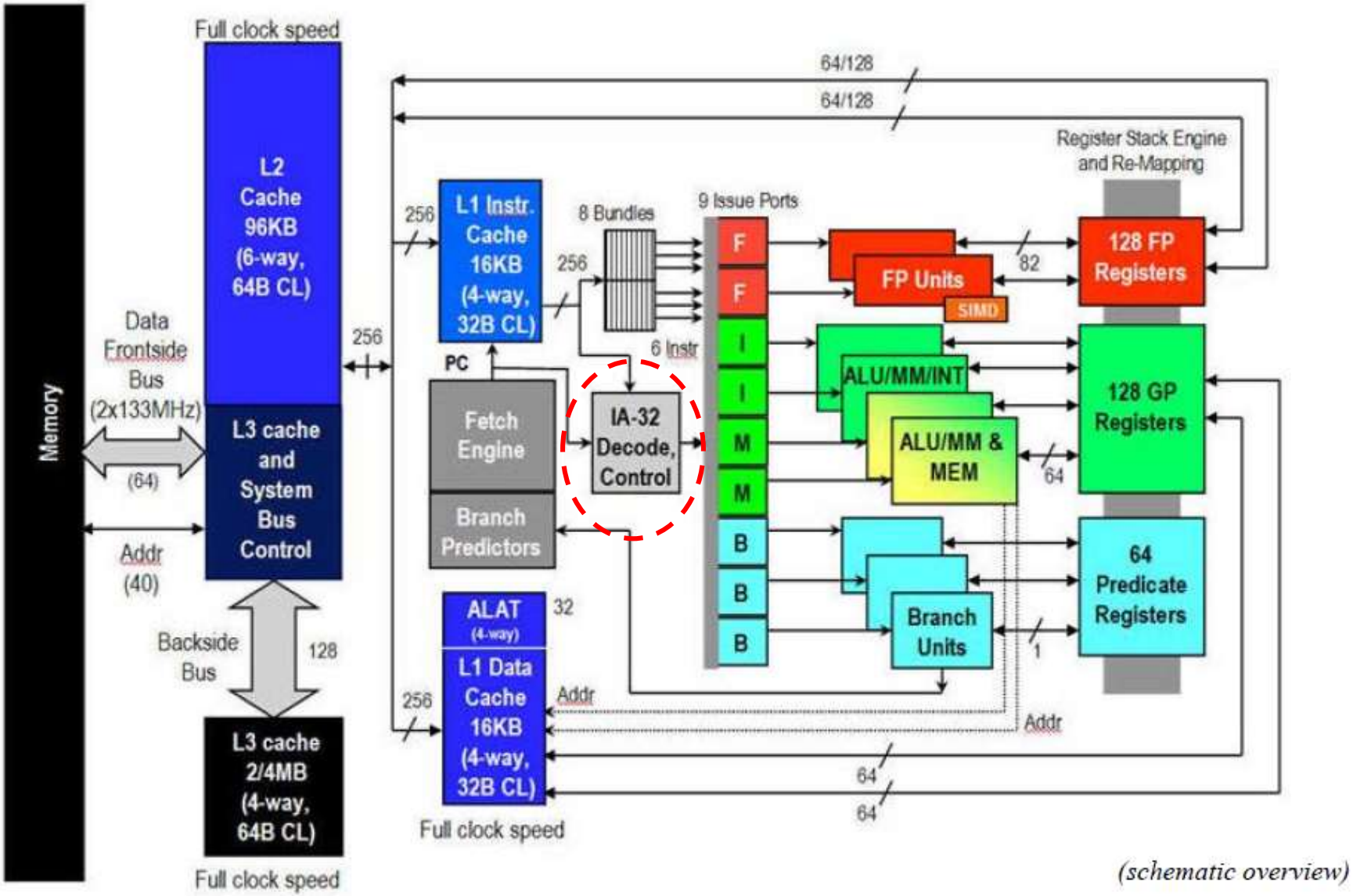


(schematic overview)

Execution of IA-32 (x86) code in the first Itanium processor (Merced)-1 [24]

- The Itanium (Merced) processor provides IA-32 (x86) compatibility by supporting the execution of the IA-32 instruction set in hardware, as indicated in the next Figure.

Execution of IA-32 (x86) code in the first Itanium processor (Merced)-2 [13]

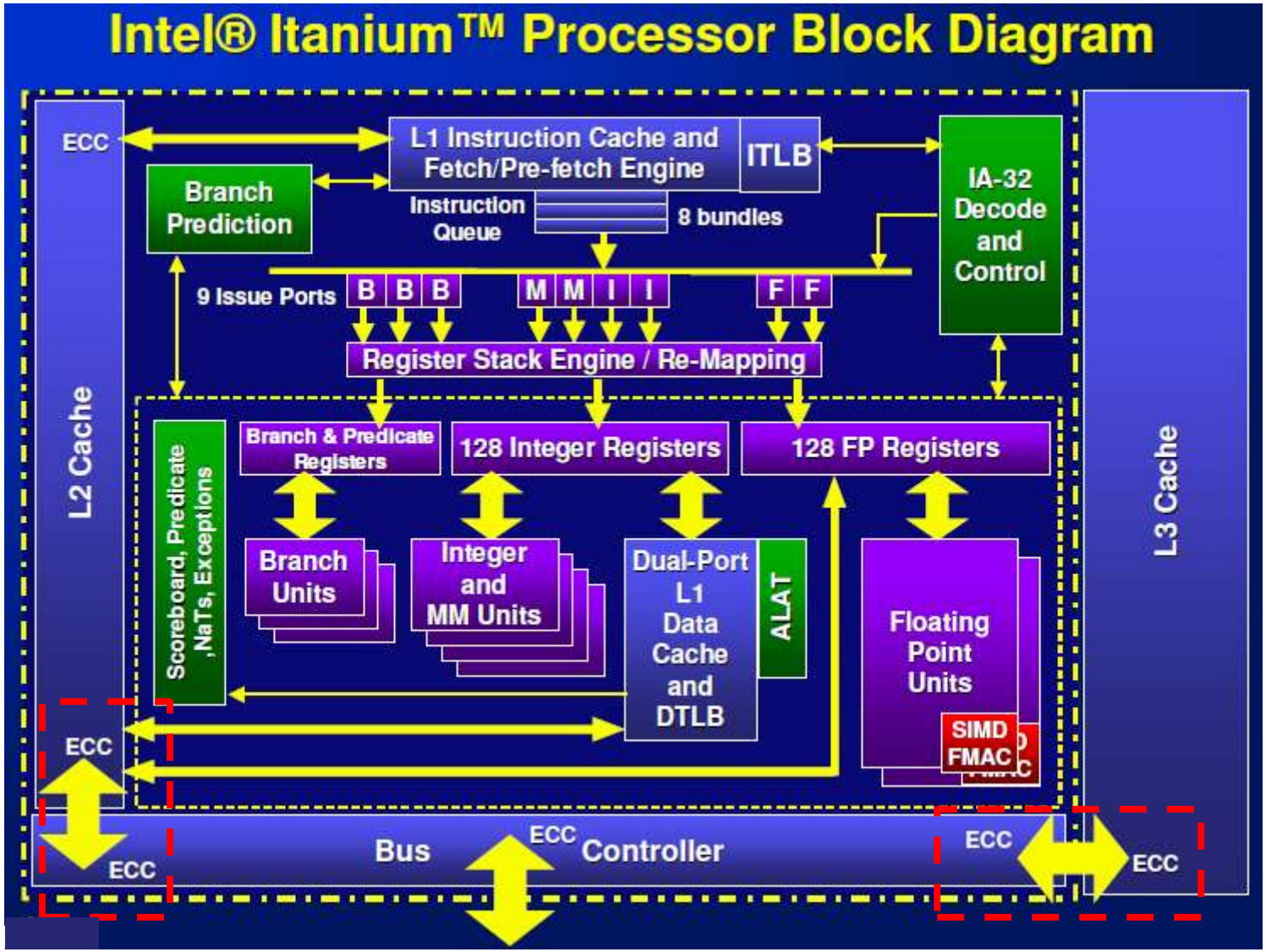


(schematic overview)

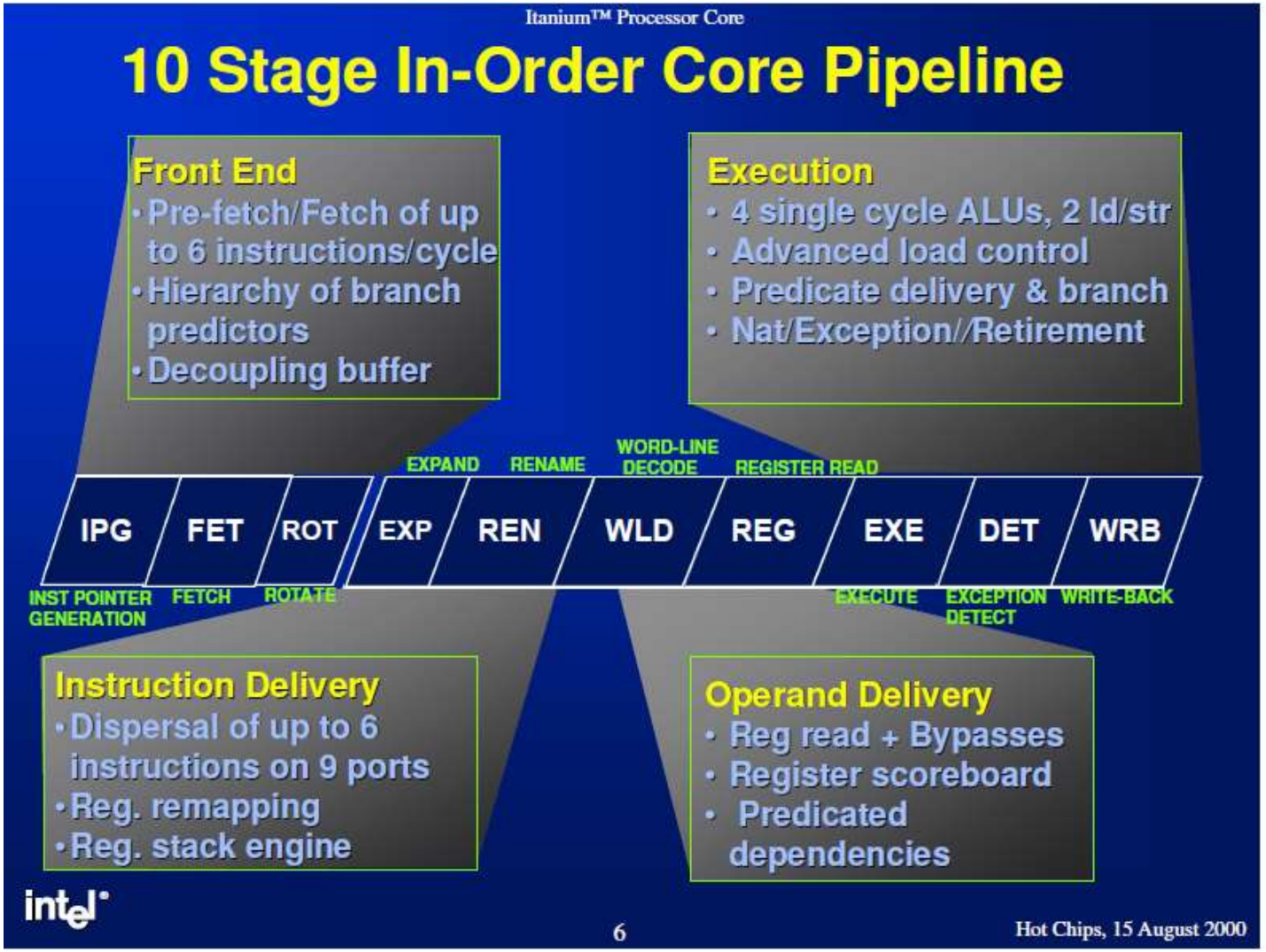
Execution of x86 code on the Itanium (Merced) core-2 [24]

- IA-32 compatibility includes support for running a mix of IA-32 and IA-64 application on an IA-64 OS, as well as IA-32 applications on an IA-32 OS.
- The IA-32 engine makes use of Merced's registers, caches and execution resources.
- Later, in 1/2003 Intel first introduced software emulation (dynamic translation) for the Itanium 2 series IA-64 processors as an option by providing a software execution layer, called the IA-32 Execution Layer (IA-32 EL), and subsequently, even removed hardware support due to efficiency issues, beginning with the Madison 9MB line in 2004, as discussed previously in Section 2.

Block diagram of the Merced processor while identifying ECC protection [14]

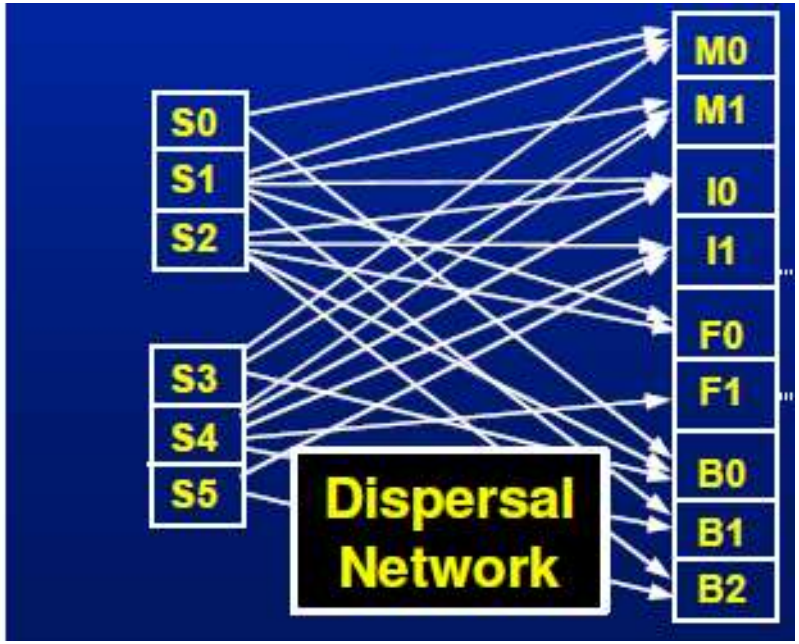


The 10-stage pipeline of the Merced processor [14]

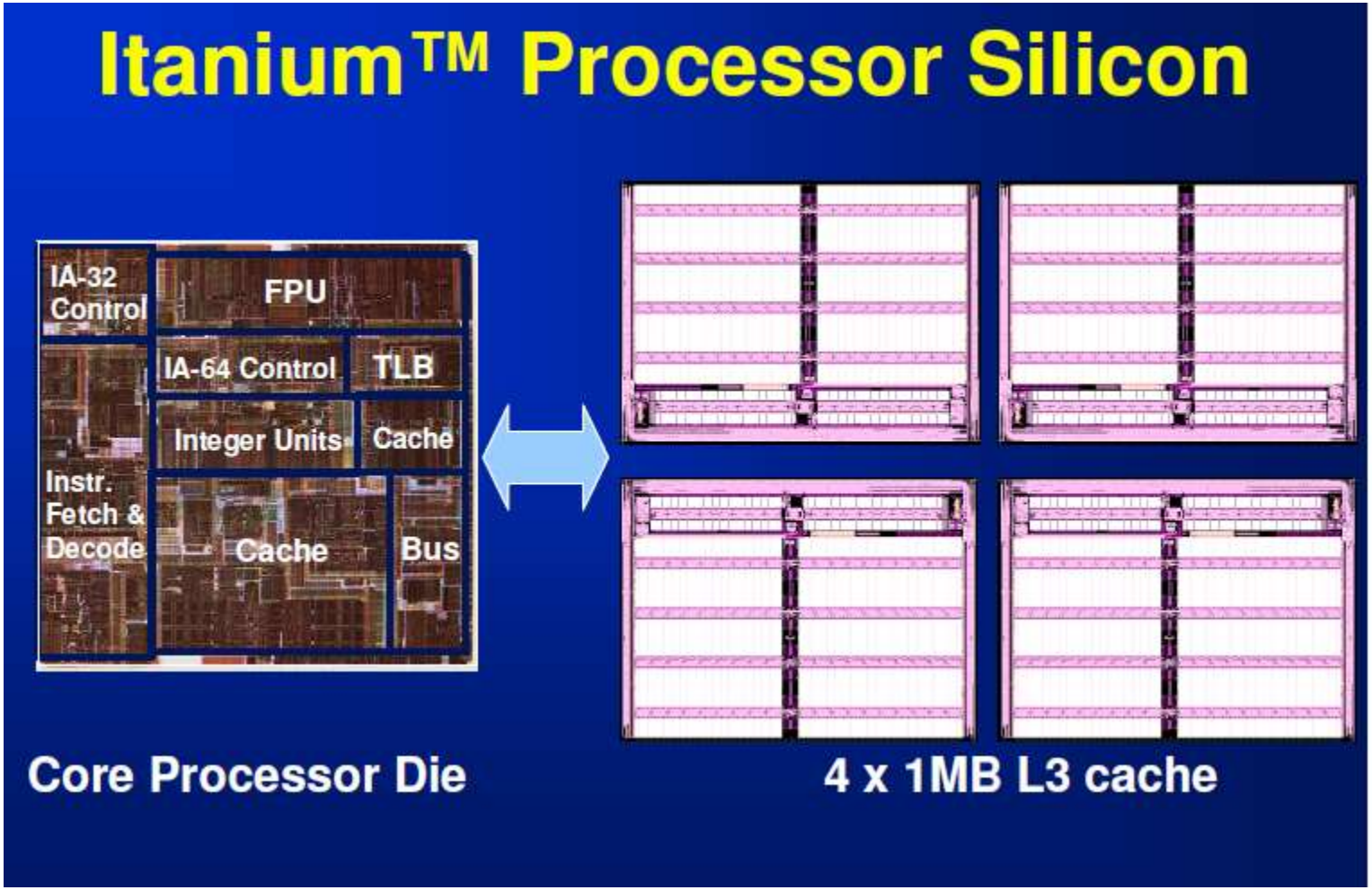


4. The Itanium line (9)

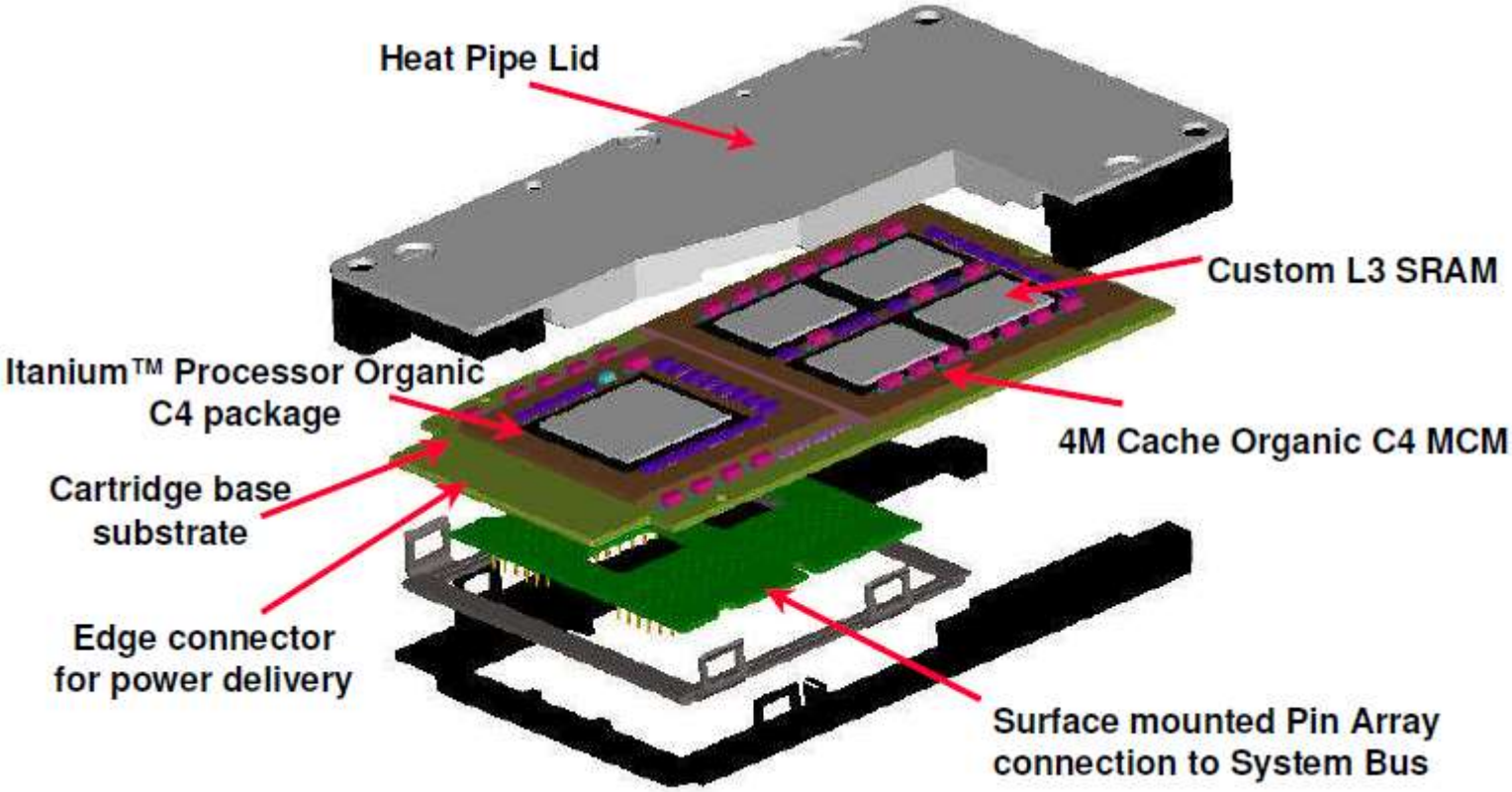
Instruction issue (dispersal) from two bundles to 9 issue ports in Merced [14



The Merced die and the four off-chip L3 cache dies [14]



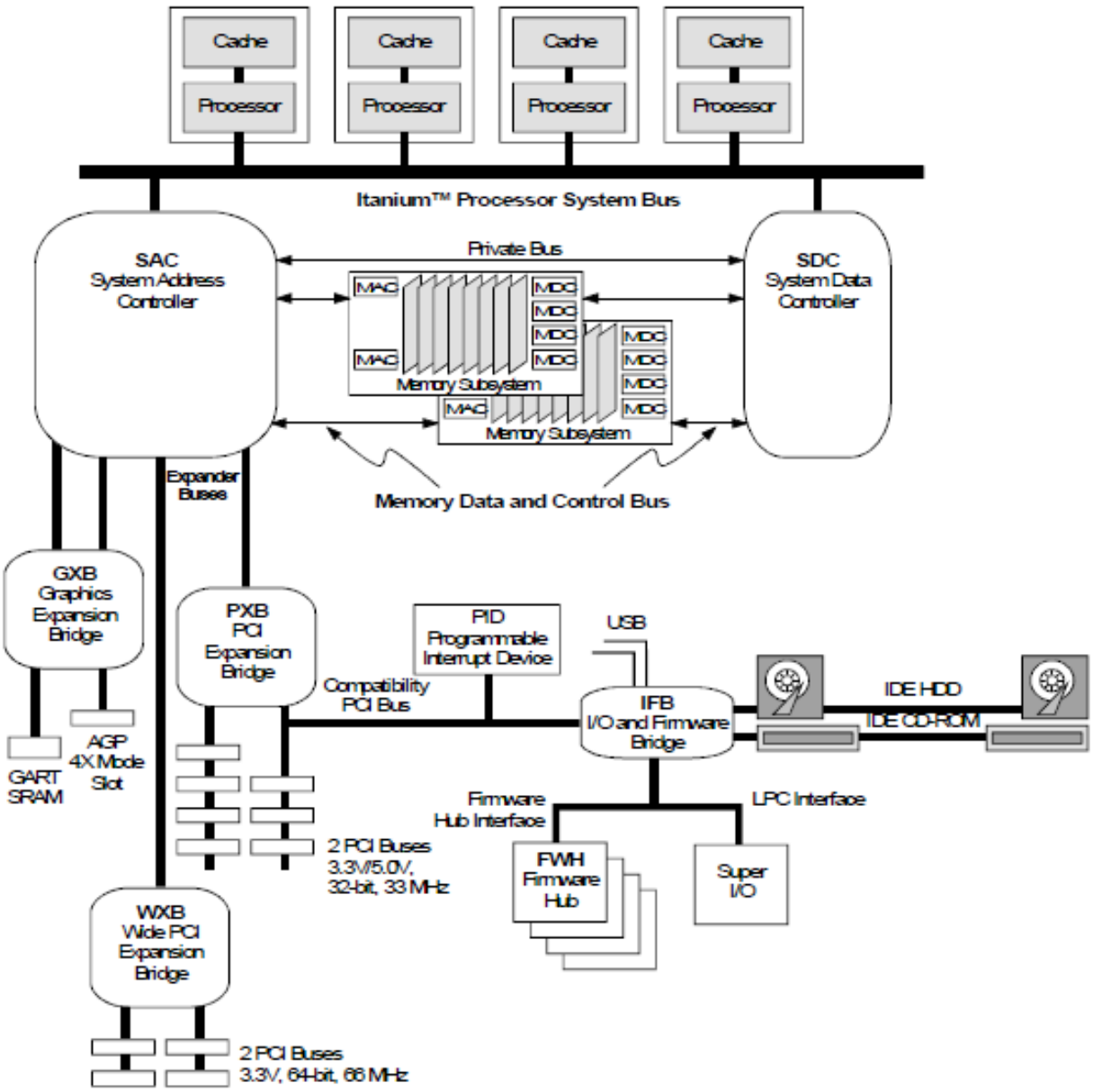
The Merced processor with a direct connected L3 [29]



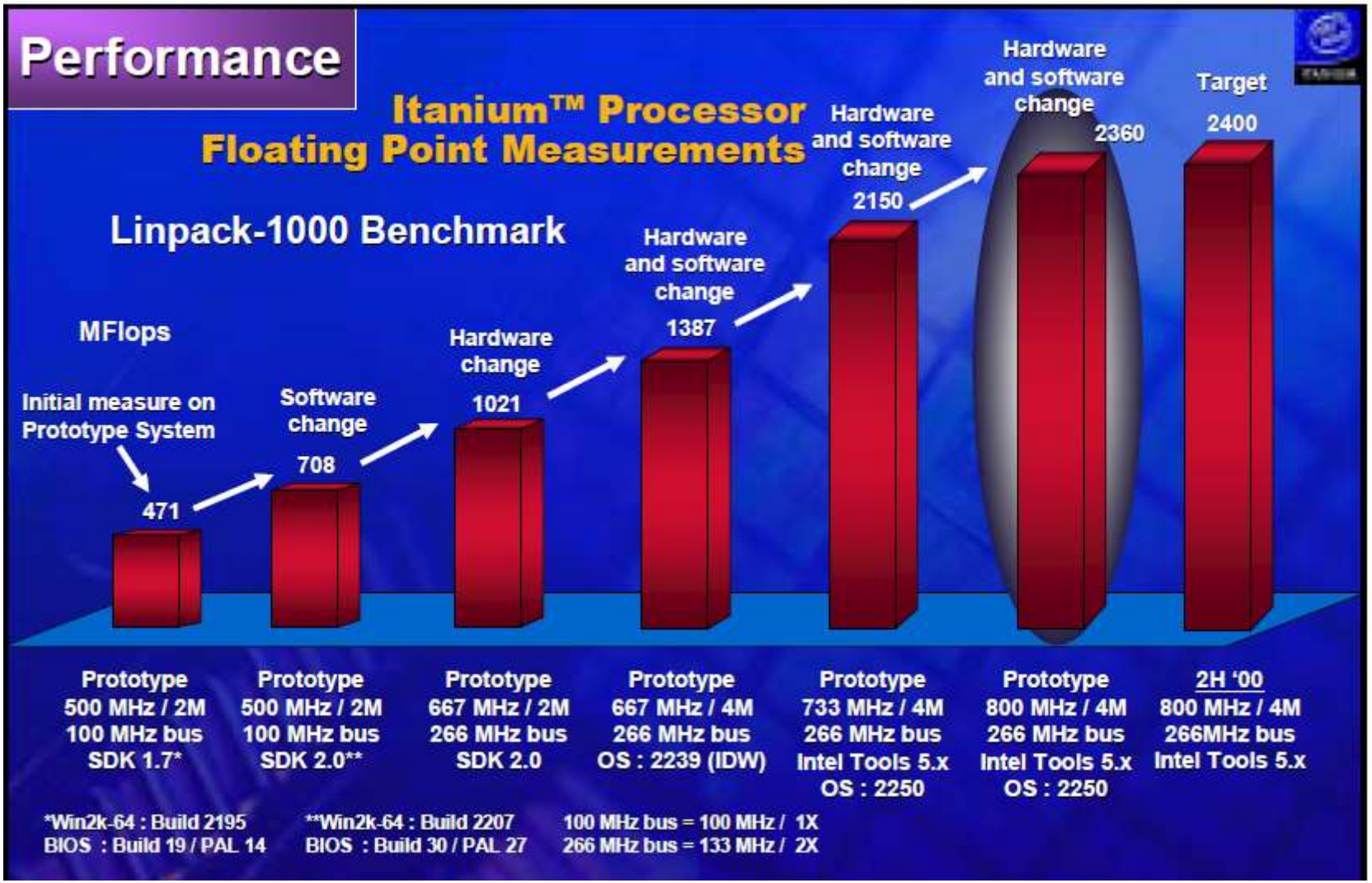
4. The Itanium line (12)

The 460GX chipset designed for the Merced [30]

The chipset consists of 10 discrete chips.

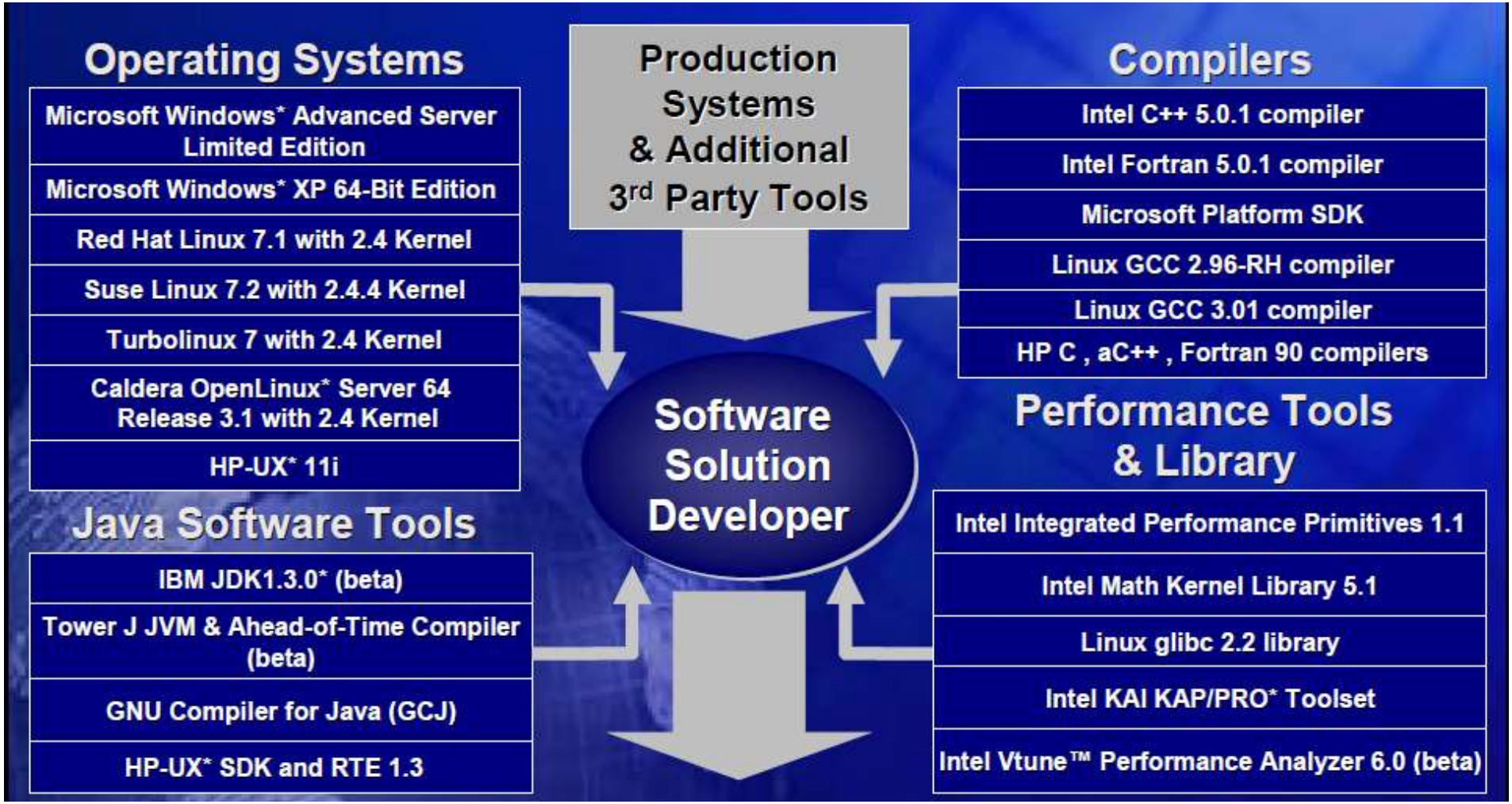


Raising the FP performance of the Merced processor during development [16



4. The Itanium line (14)

Itanium processor software development environment available in 4/2002 [1

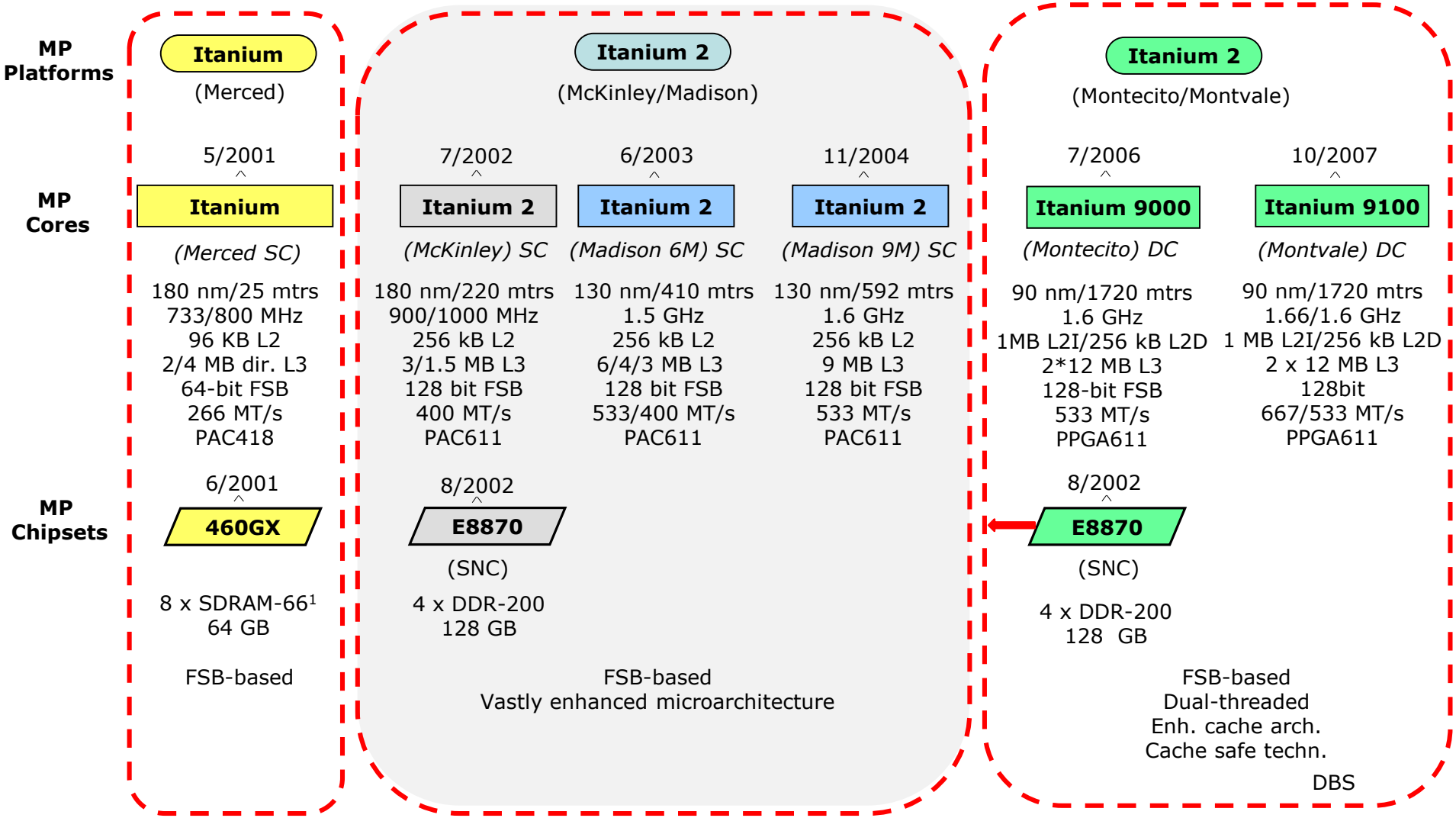


5. The Itanium 2 McKinley/Madison platform

- 5.1 Overview of the Itanium 2 (McKinley/Madison) platform
- 5.2 The Itanium 2 McKinley line
- 5.3 The Itanium 2 Madison 6 MB L3 line
- 5.4 The Itanium 2 Madison line with 9 MB L3

5.1 Overview of the Itanium 2 (McKinley/Madison) platform

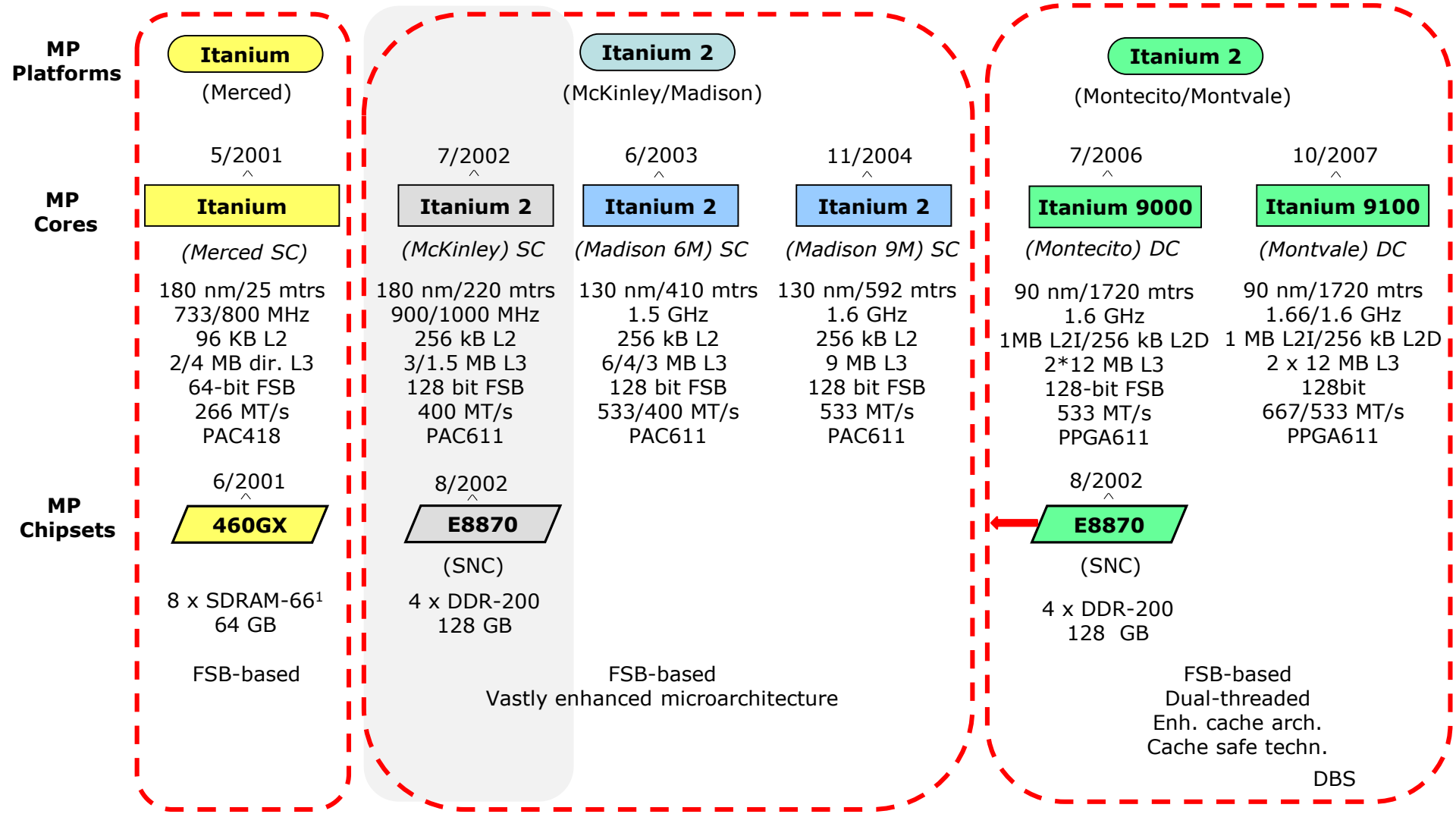
5.1 Overview of the Itanium 2 McKinley/Madison platform



[1]: Special memory cards are used SNC: Scalable Node Controller

5.2 The Itanium 2 McKinley line

5.2 The Itanium 2 McKinley line-1



[1]: Special memory cards are used SNC: Scalable Node Controller

5.2 The Itanium 2 McKinley line-2

- Introduced in 7/2002
- Feature size: 180 nm

Main enhancements of McKinley

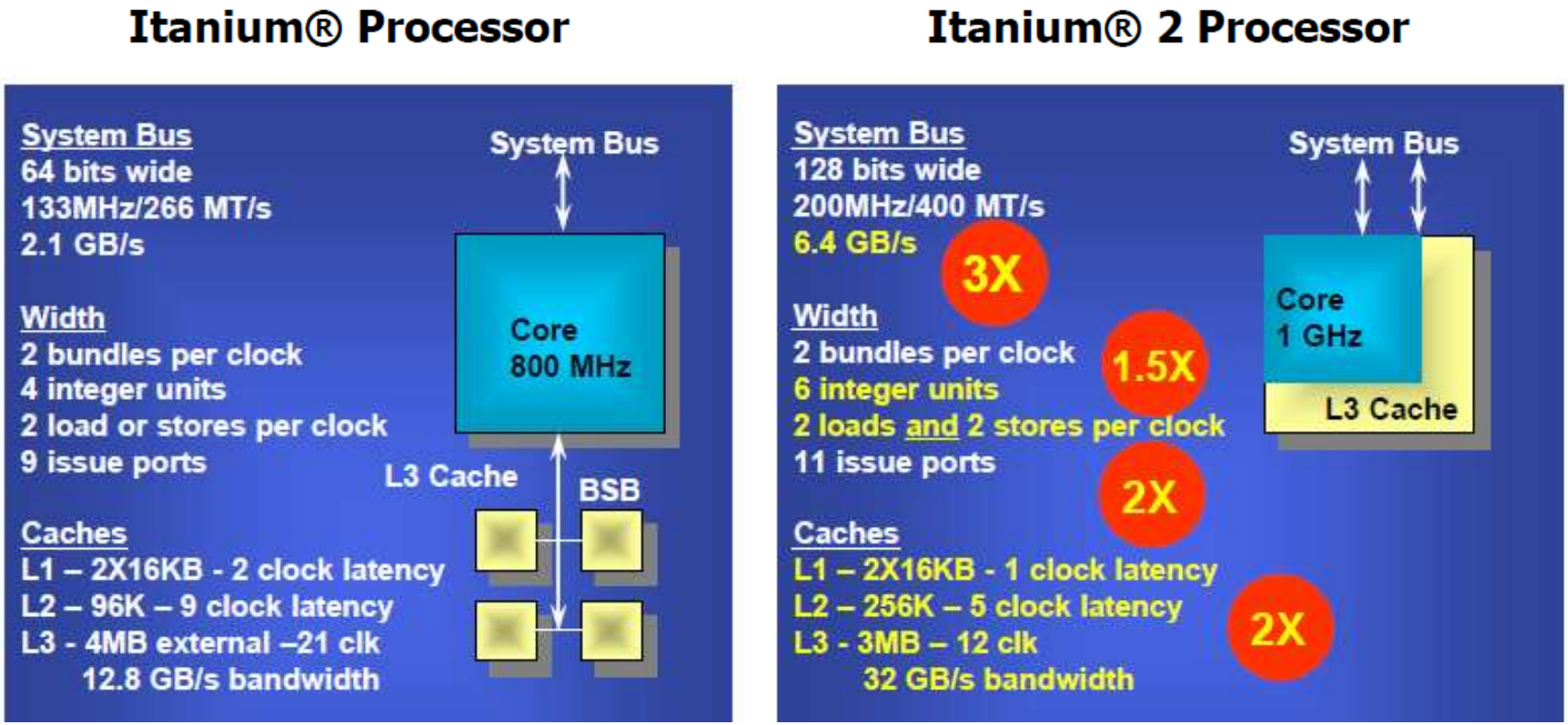
a) Extended address space
b) Vastly enhanced microarchitecture
c) New enhanced chipset
    - allowing to build up to 4 x 4 processor SMPs
    - supporting snoop filter based cache coherency protocol.

5.2 The Itanium 2 McKinley line (3)

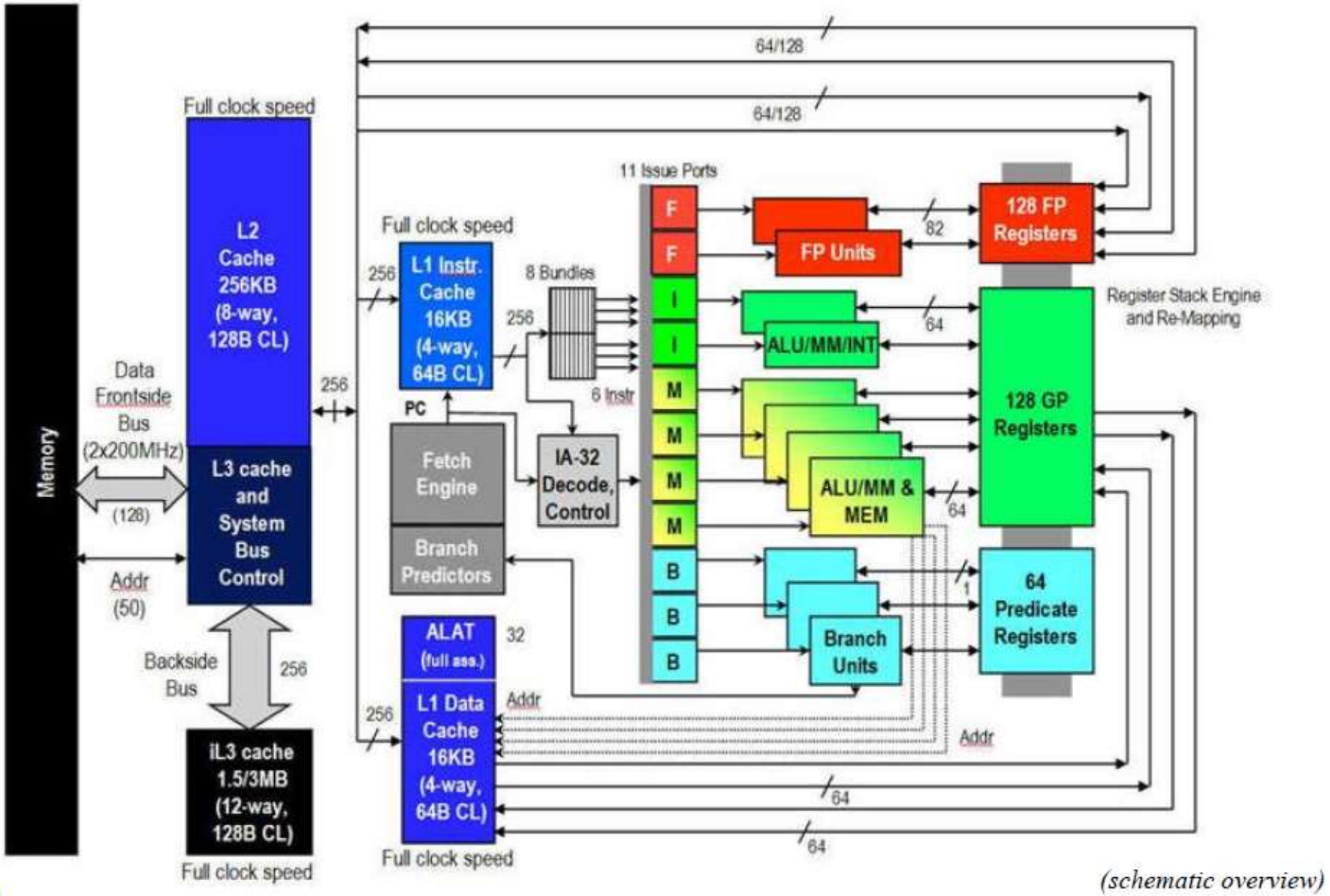
a) Extended address space [13]

Itanium® Processor	Itanium® 2 Processor
Addressing 44 bit physical addressing 50 bit virtual addressing Maximum page size of 256MB	Addressing 50 bit physical addressing 64 bit virtual addressing Maximum page size of 4GB

b) Vastly enhanced microarchitecture [13]



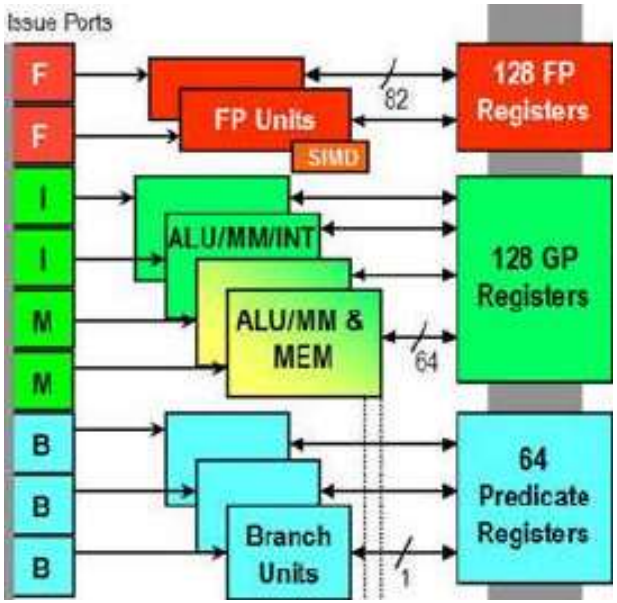
Block diagram of the Itanium 2 (McKinley) processor [13]



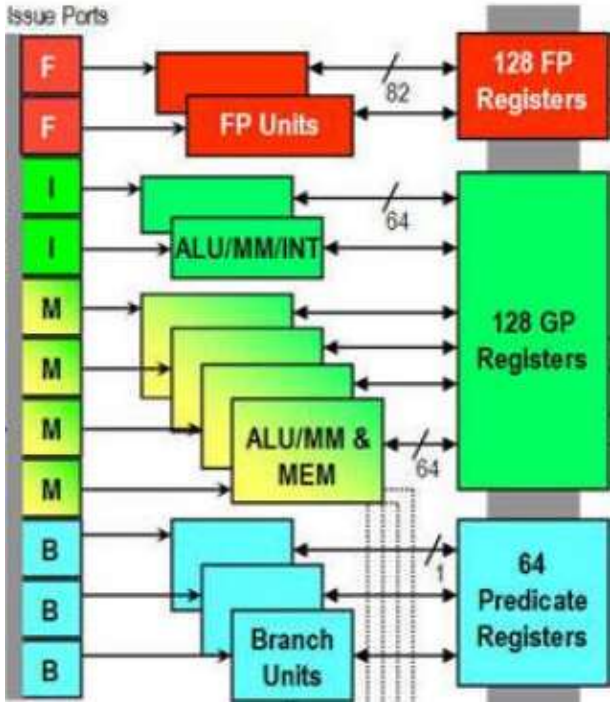
5.2 The Itanium 2 McKinley line (6)

Increasing the number of available issue ports (based on [13])

Itanium



Itanium 2



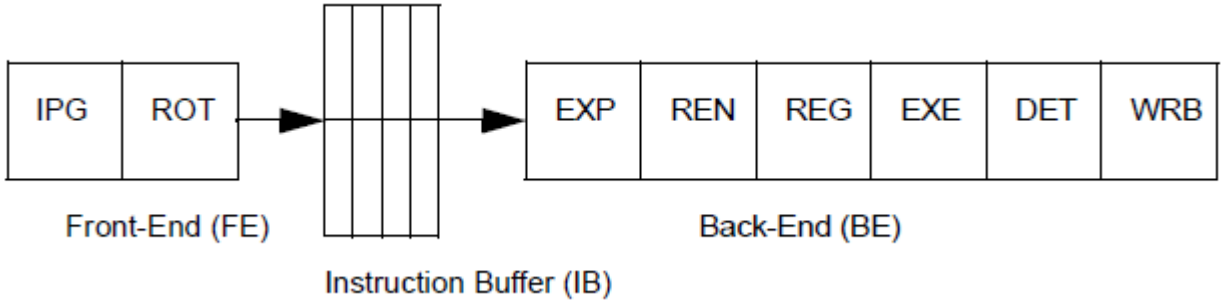
The first Itanium (Merced) only had two memory pipelines, which was a tremendous problem for performance.

One of the largest improvements in McKinley was adding two memory pipelines, which increased ILP significantly [31].

5.2 The Itanium 2 McKinley line (7)

Reducing the pipeline depth from 10 to 8 [32]

The Itanium 2 pipeline



The core pipeline consists of eight stages:

IPG: Instruction pointer generation

ROT: instruction rotation

EXP: Instruction template decode, expand, and disperse

REN: Rename (for register stack and rotating registers) and decode

REG: Register file read

EXE: ALU execution

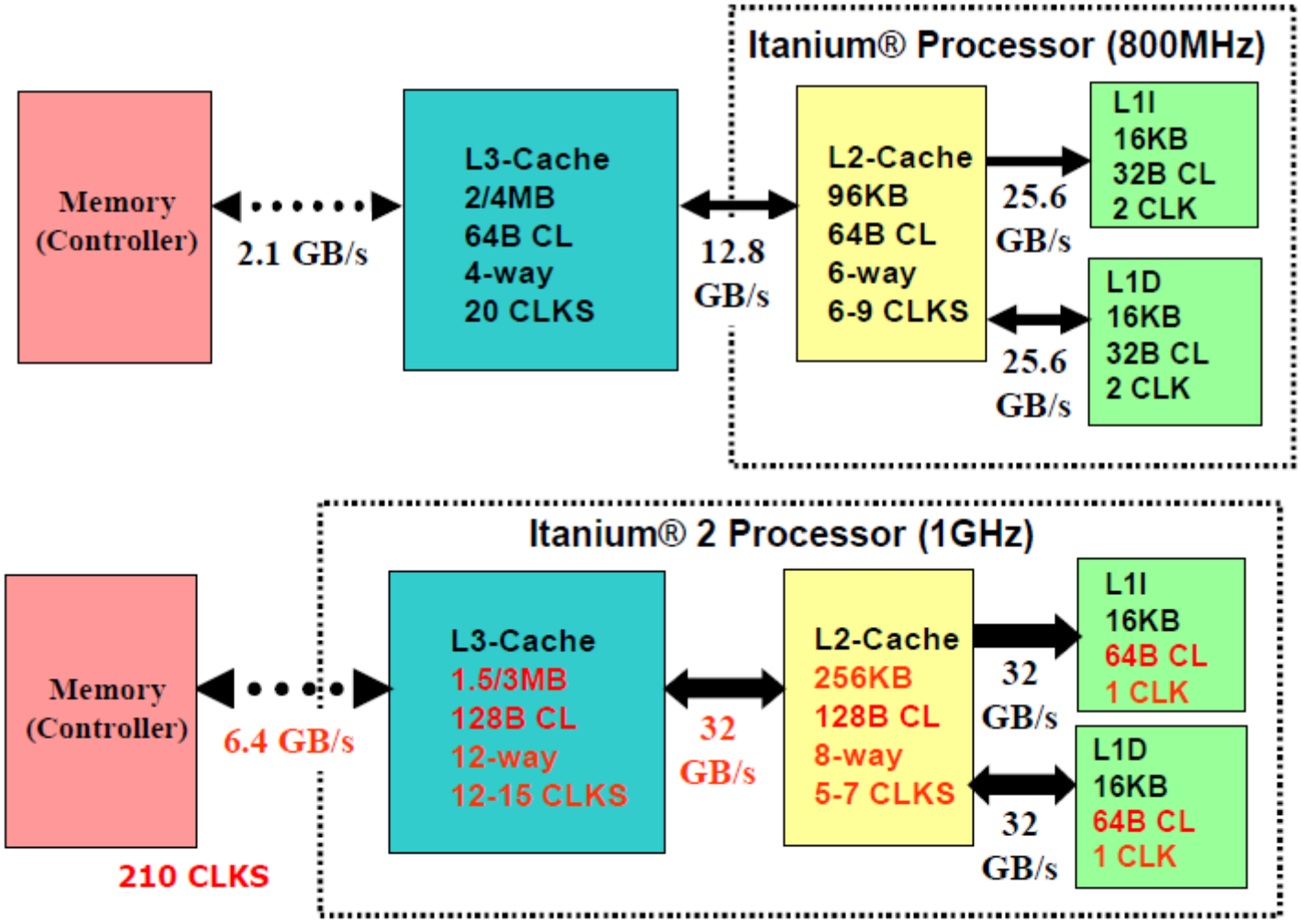
DET: Last stage for exception detection

WRB: Write back

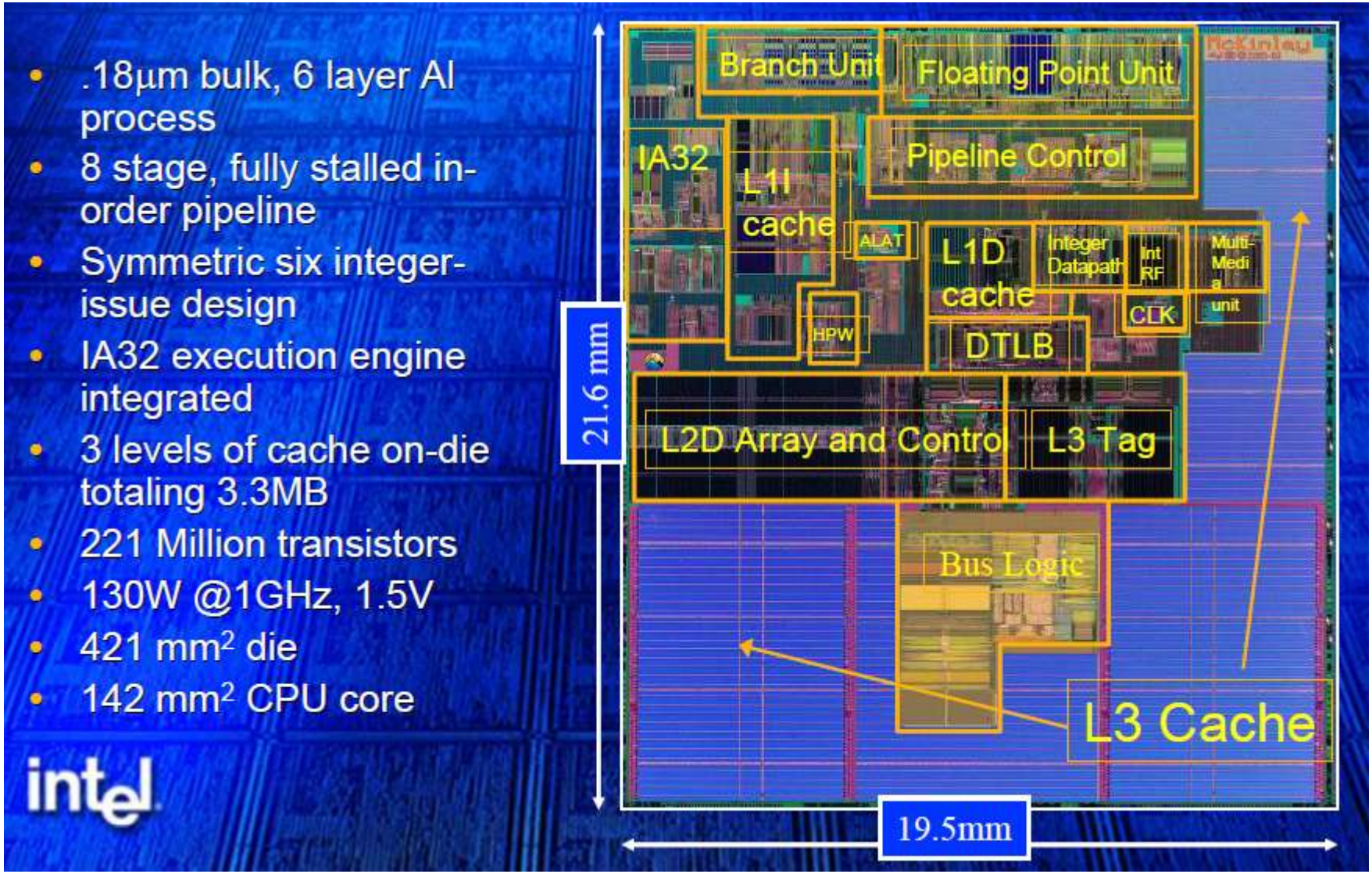
Note that Itanium 2 processors have a shorter pipeline then Merced as two stages of Merced's pipeline were eliminated (the FET (Fetch) and the WLD (World Line Decode) stages).

5.2 The Itanium 2 McKinley line (8)

Contrasting the cache hierarchies of Itanium and Itanium 2 processors [13]

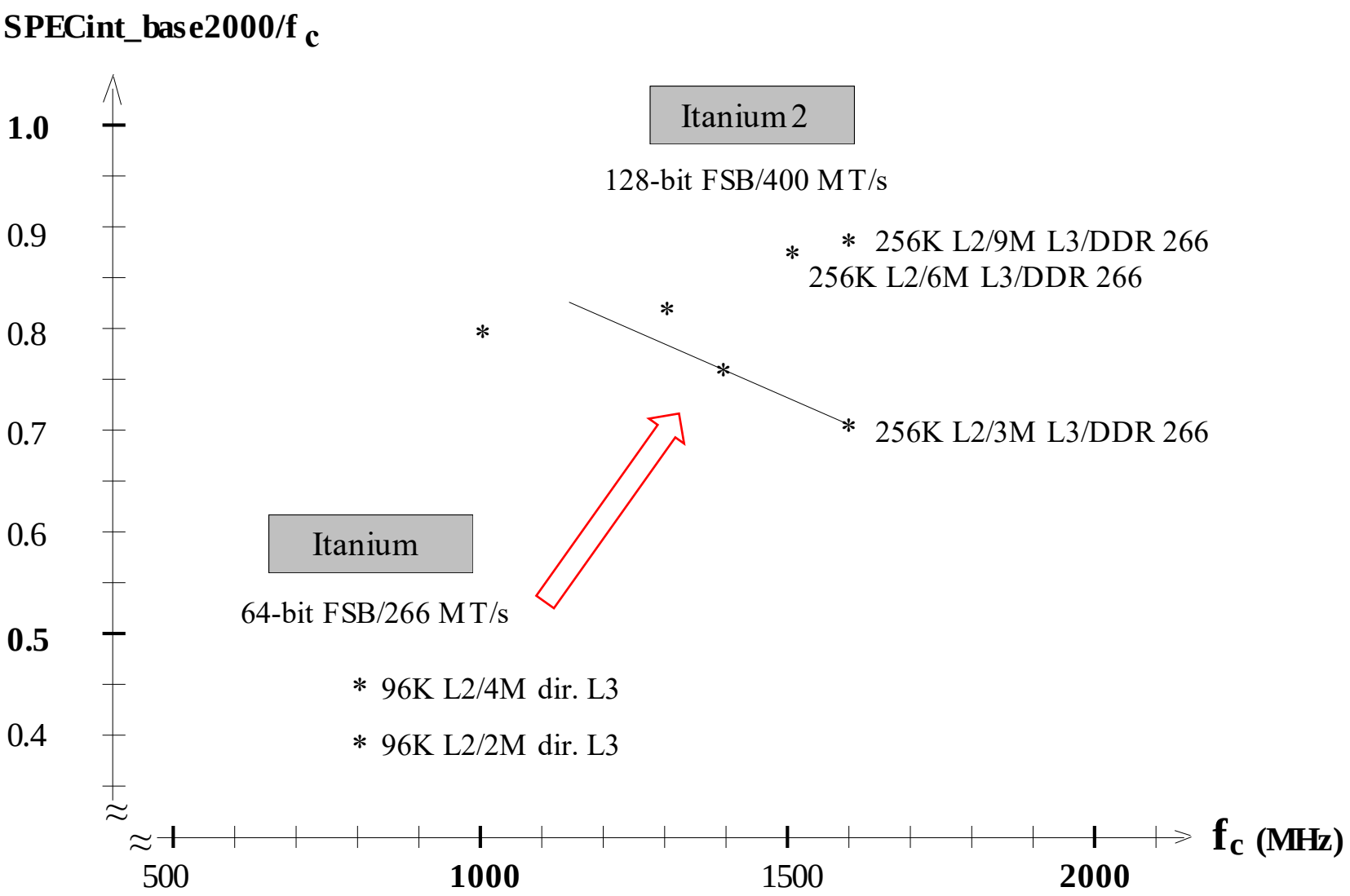


Itanium 2 McKinley's floor plan [33]



5.2 The Itanium 2 McKinley line (10)

The increased efficiency of the Itanium 2 (McKinley) processor over Merced



c) New enhanced chipset

Itanium 2 processors are supported by the E8870 chipset.

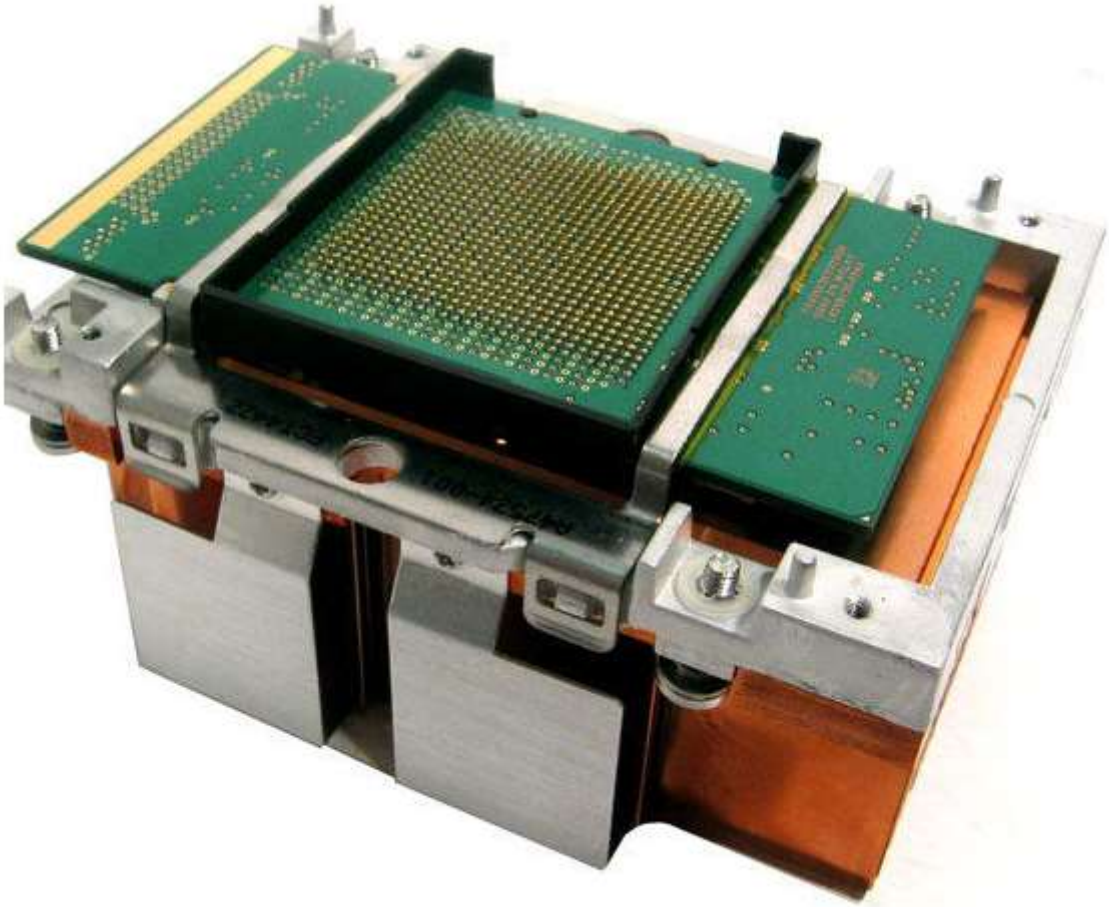
Key features of the new chipset

- allowing to build up to 16 processor SMMs
  (Shared Memory Multiprocessors)
- supporting snoop filter based cache coherency protocol,

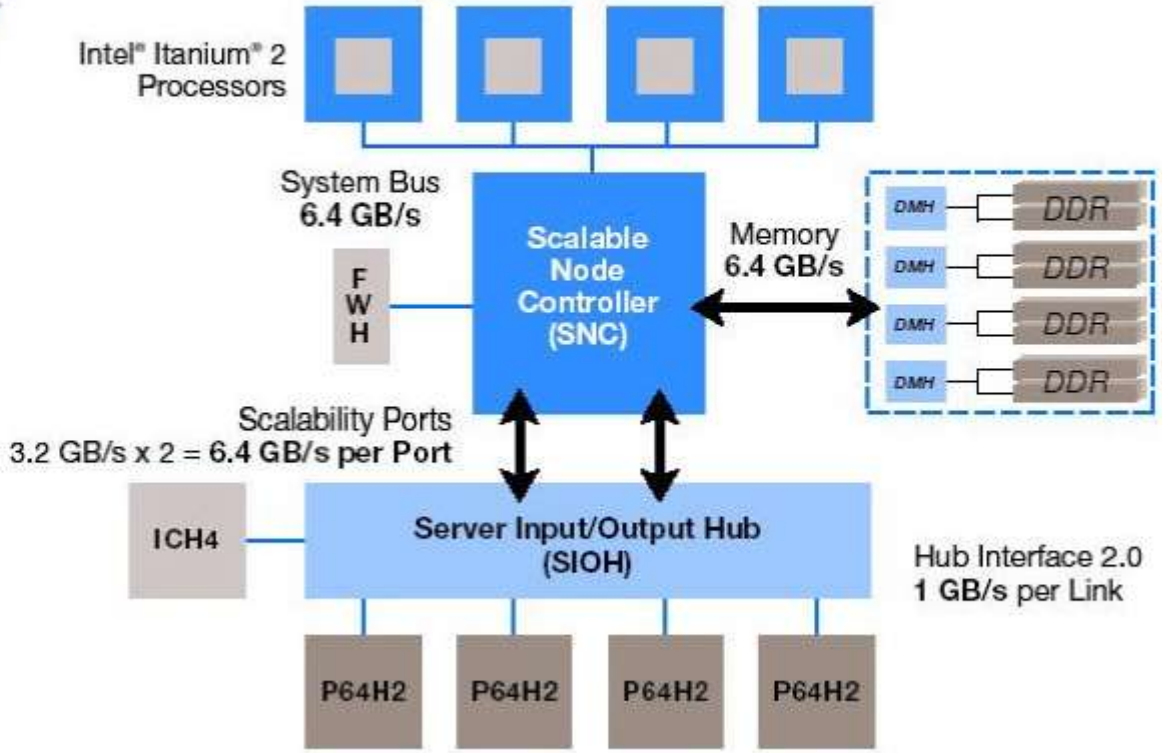
as briefly discussed subsequently.

5.2 The Itanium 2 McKinley line (12)

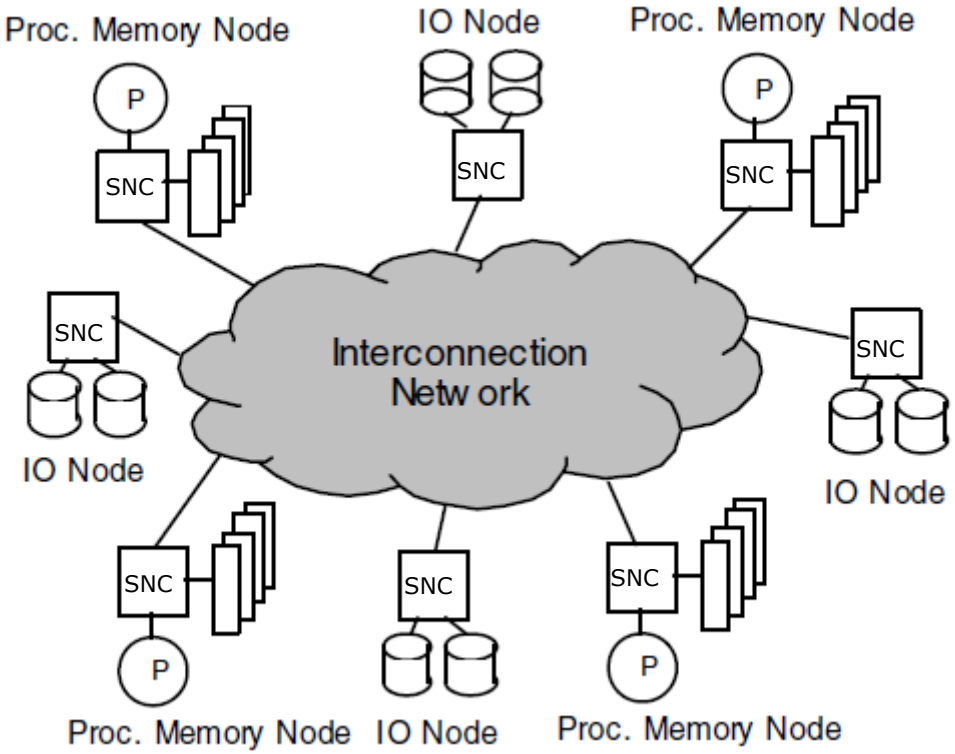
The PAC611 socket for the Itanium 2 platform with cooler [34]



Example 1: E8870 chipset based MP server [13]



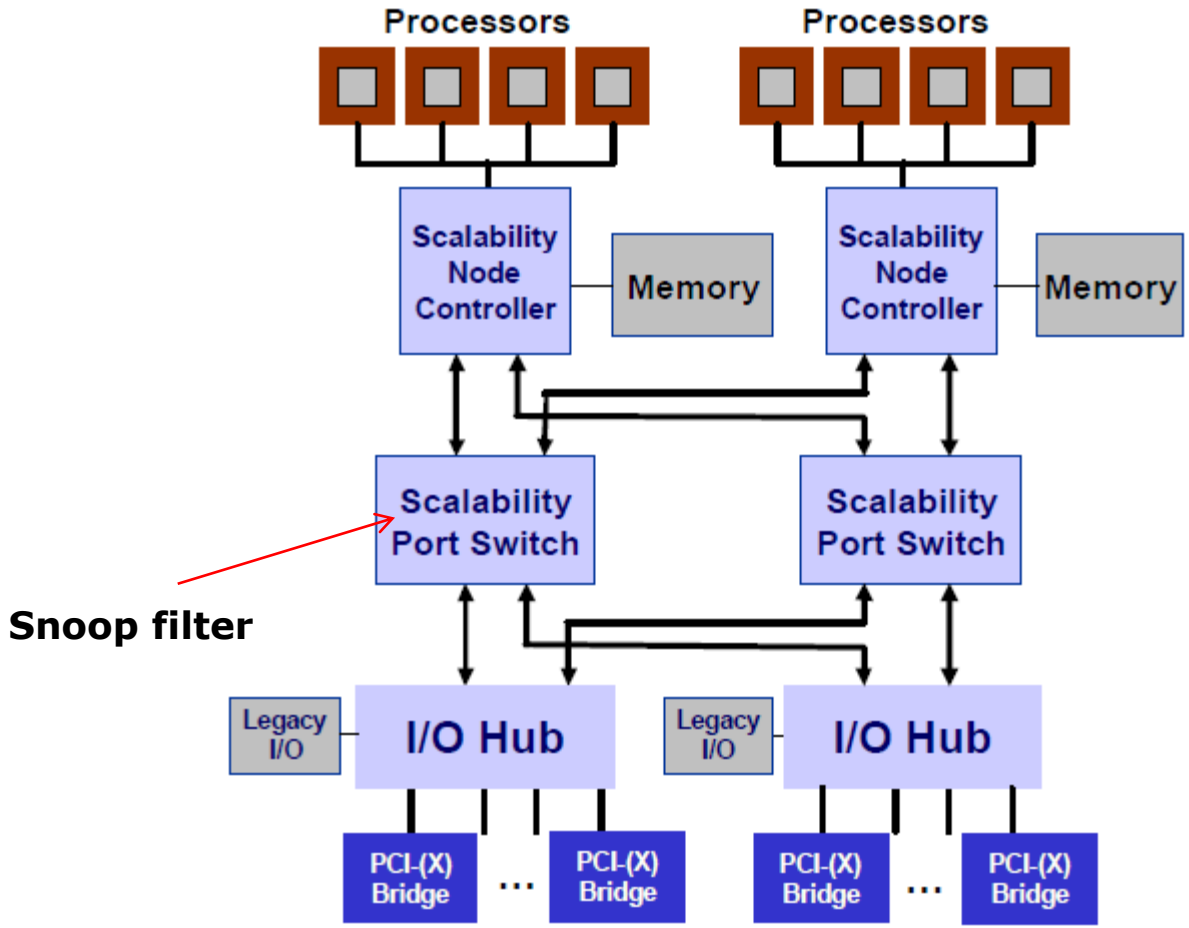
Principle of connecting Processor Memory nodes and I/O Nodes by a coherent interconnection network [35]



SNC: Scalability Node Controller (E8870)

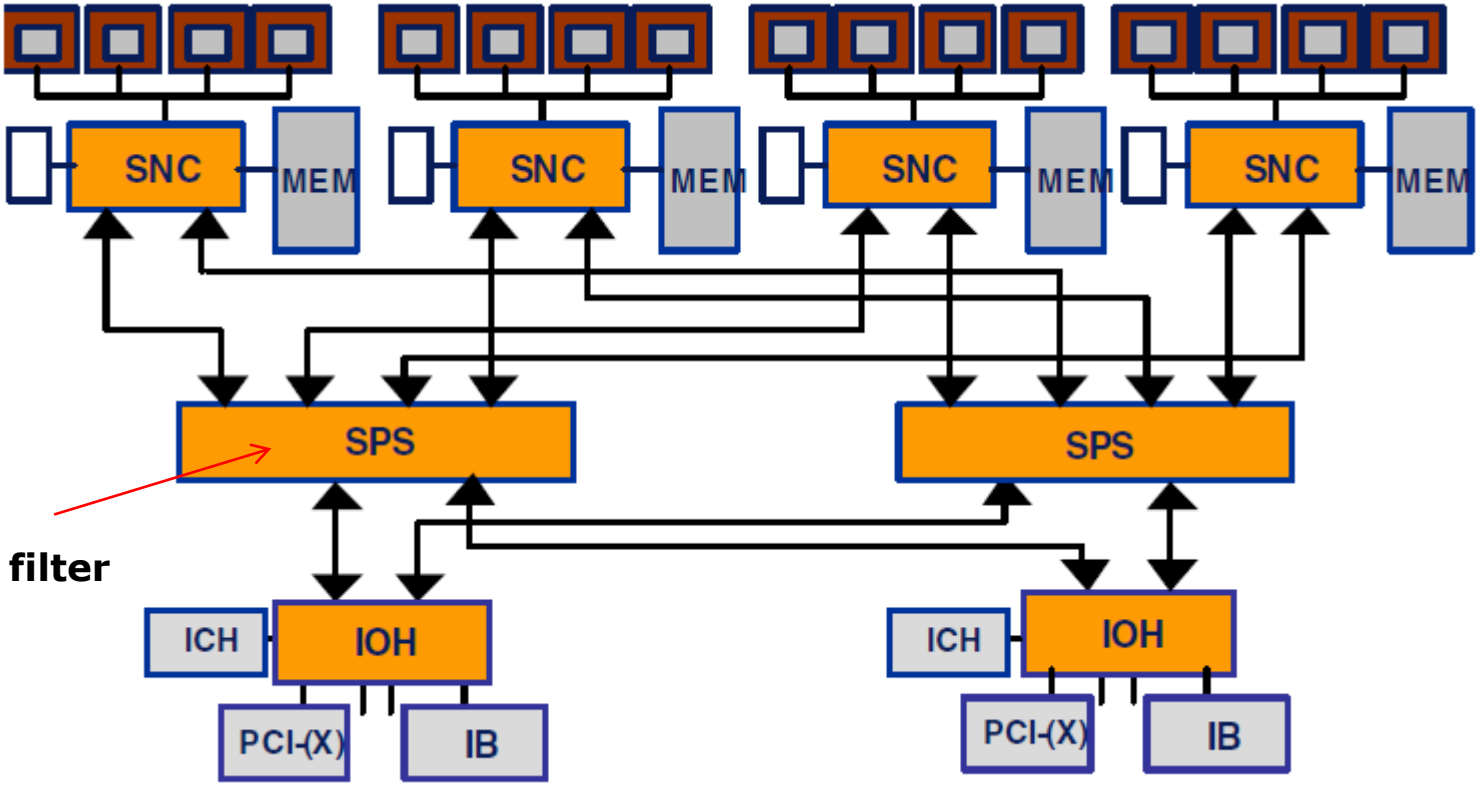
Scalability Port (SP)

Example 2: E8870 chipset based 8-processor SMM [13]



SNC: Scalability Node Controller (E8870)
SPS: Scalability Port Switch (E9870)
SMM: Shared Memory Multiprocessor

Example 3: E8870 chipset based 16 processor SMM [35]



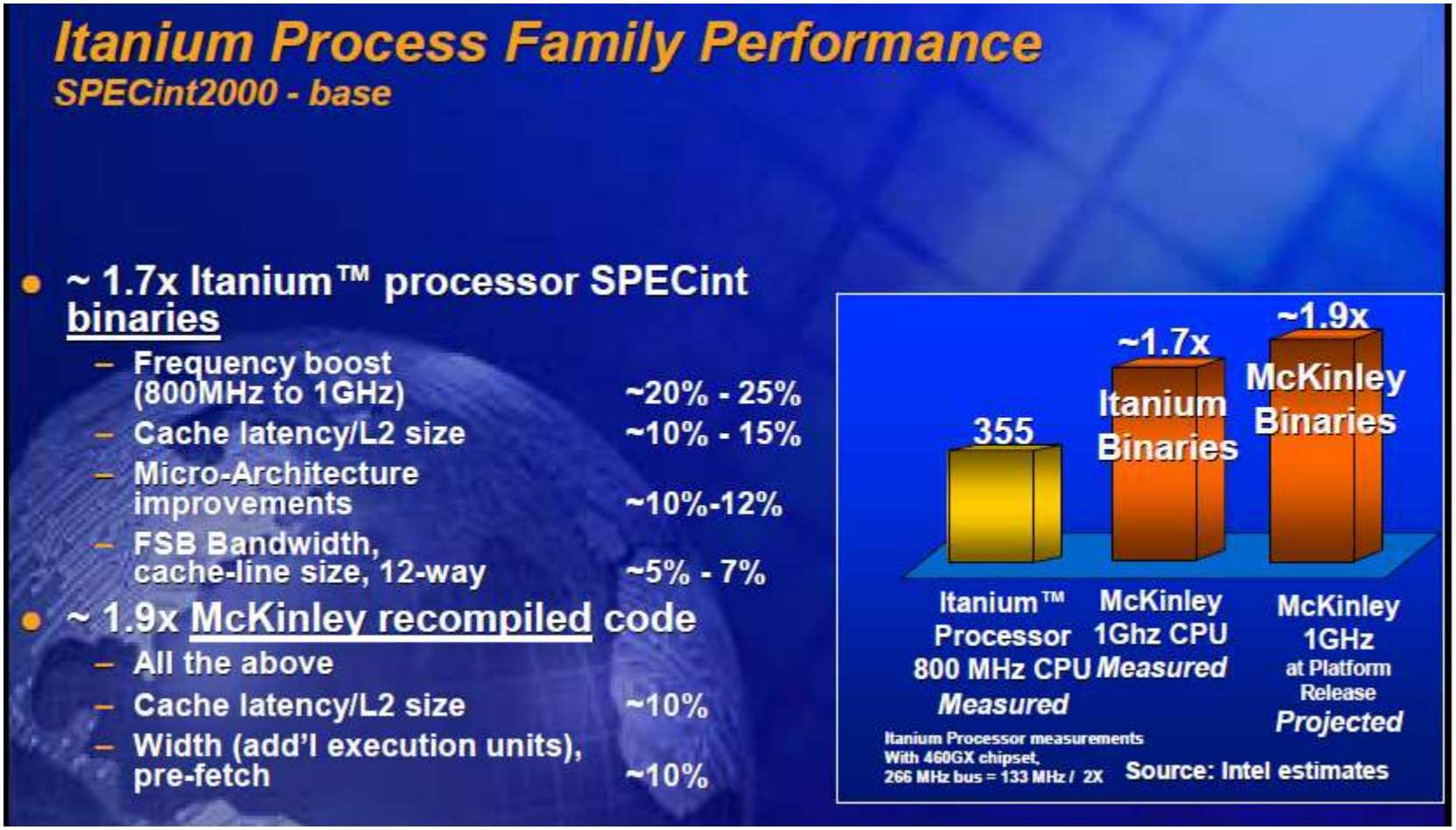
SNC: Scalability Node Controller (E8870)Switch
SPS: Scalability Port Switch (E9870)
SMM: Shared Memory Multiprocessor

Maintaining cache coherency

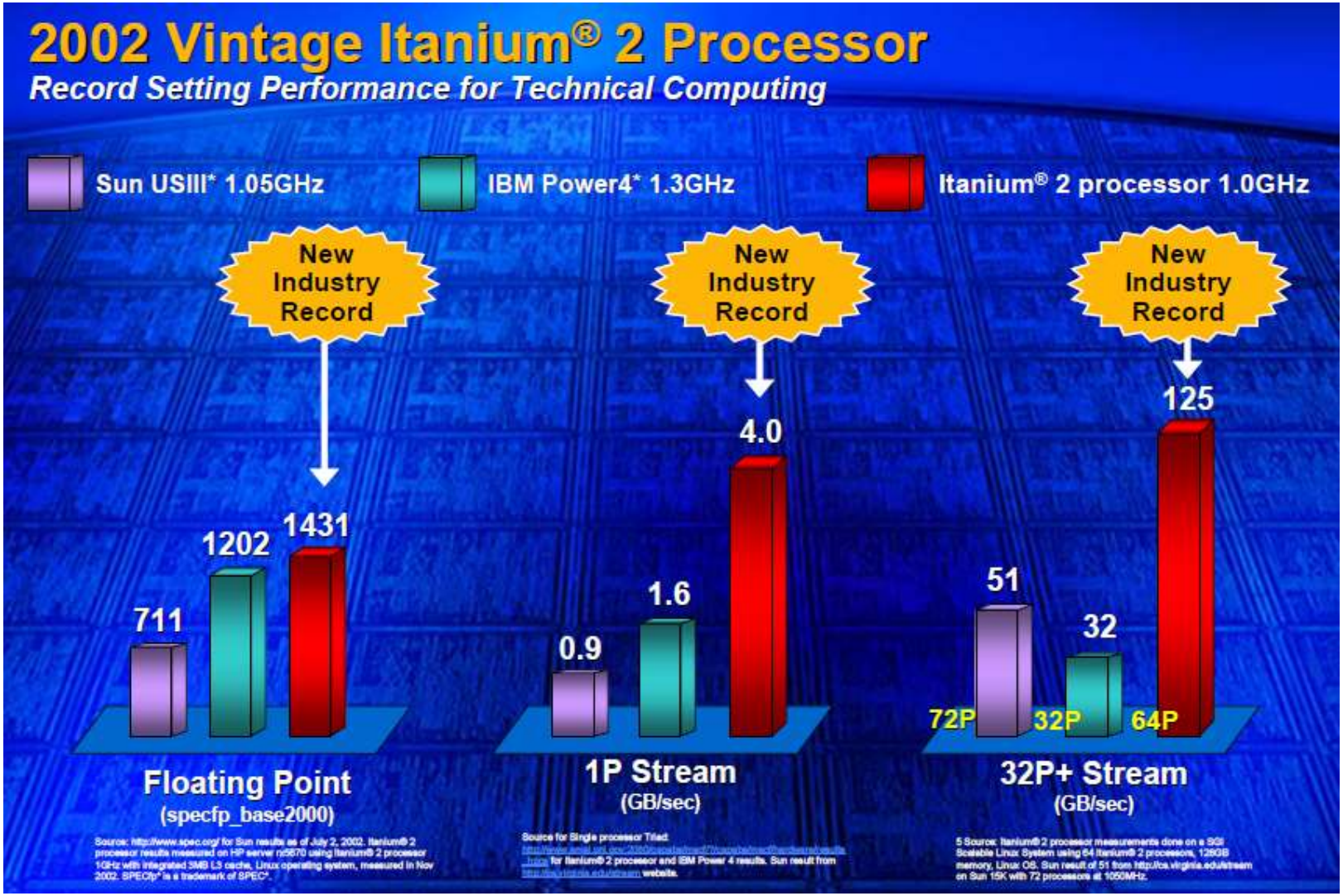
The E8870/E9870 chipset makes use of a snoop filter supported cache coherency protocol to implement multi node (n x 4) shared memory systems (SMMs).

For a comprehensive description of Intel's cache coherency solution for Itanium 2 systems see [35].

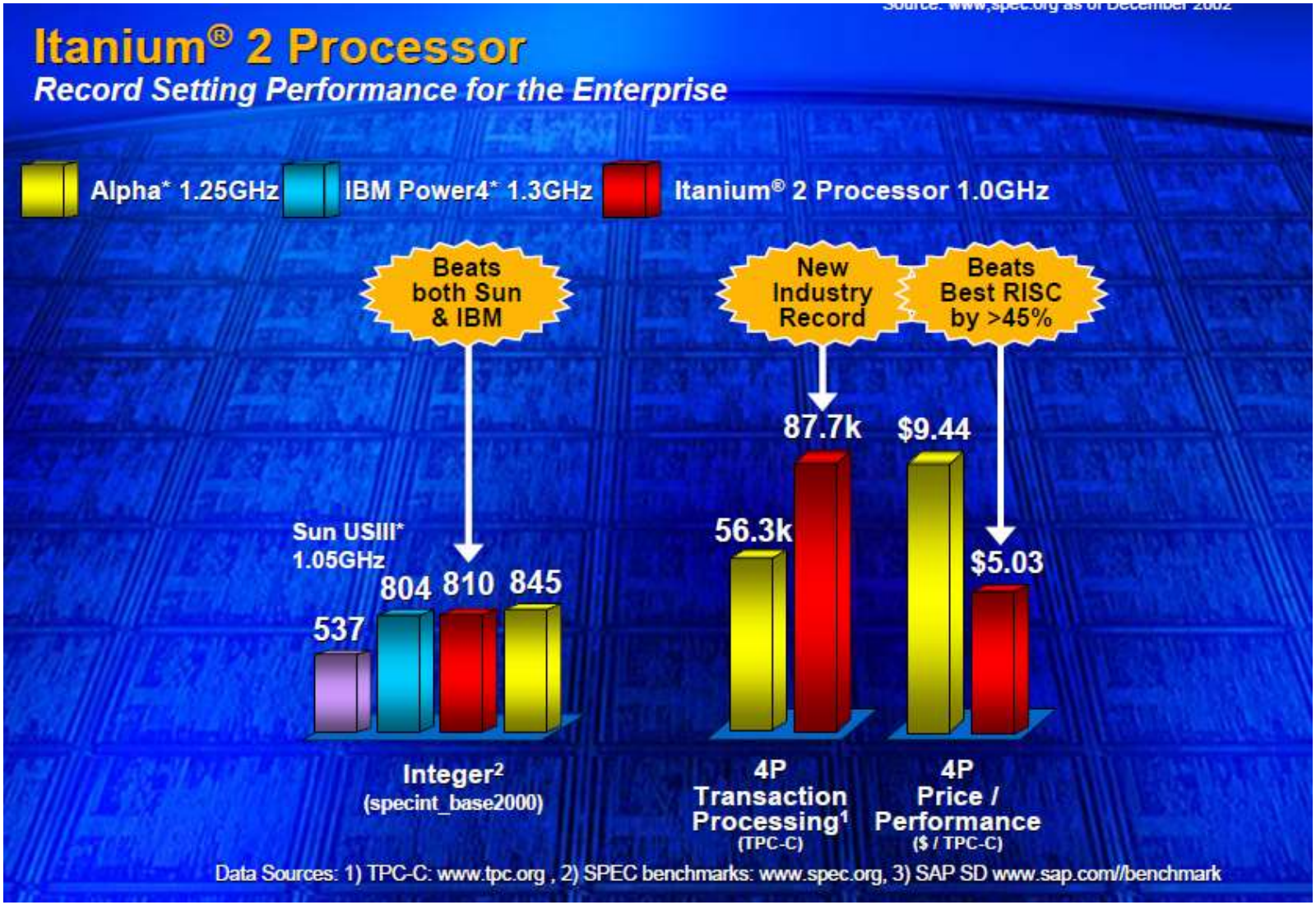
Itanium 2 McKinley's performance vs. the original Itanium performance [11]



Performance achievements of Itanium 2 McKinley for technical computing [33

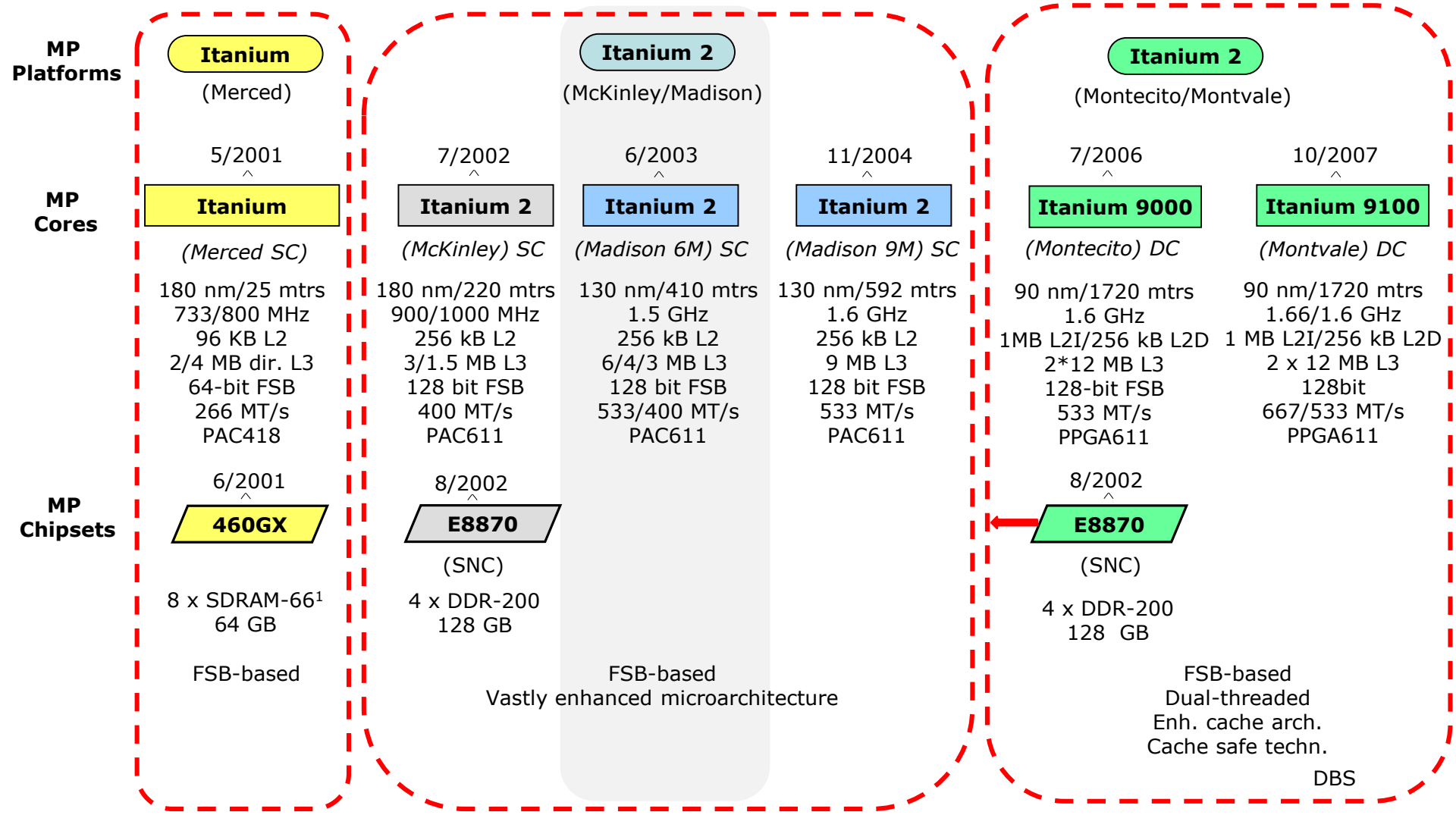


Performance achievements of Itanium 2 McKinley for enterprise computing [3



5.3 The Itanium 2 Madison 6 MB L3 line

5.3 The Itanium 2 Madison 6M line-1



[1]: Special memory cards are used SNC: Scalable Node Controller

5.3 The Itanium 2 Madison 6 MB L3 line (2)

The Itanium 2 Madison 6 M line

Introduction: 6/2003
Introduced models

- 1.5 GHz 6 MB L3
- 1.4 GHz 4 MB L3
- 1.3 GHz 3 MB L3

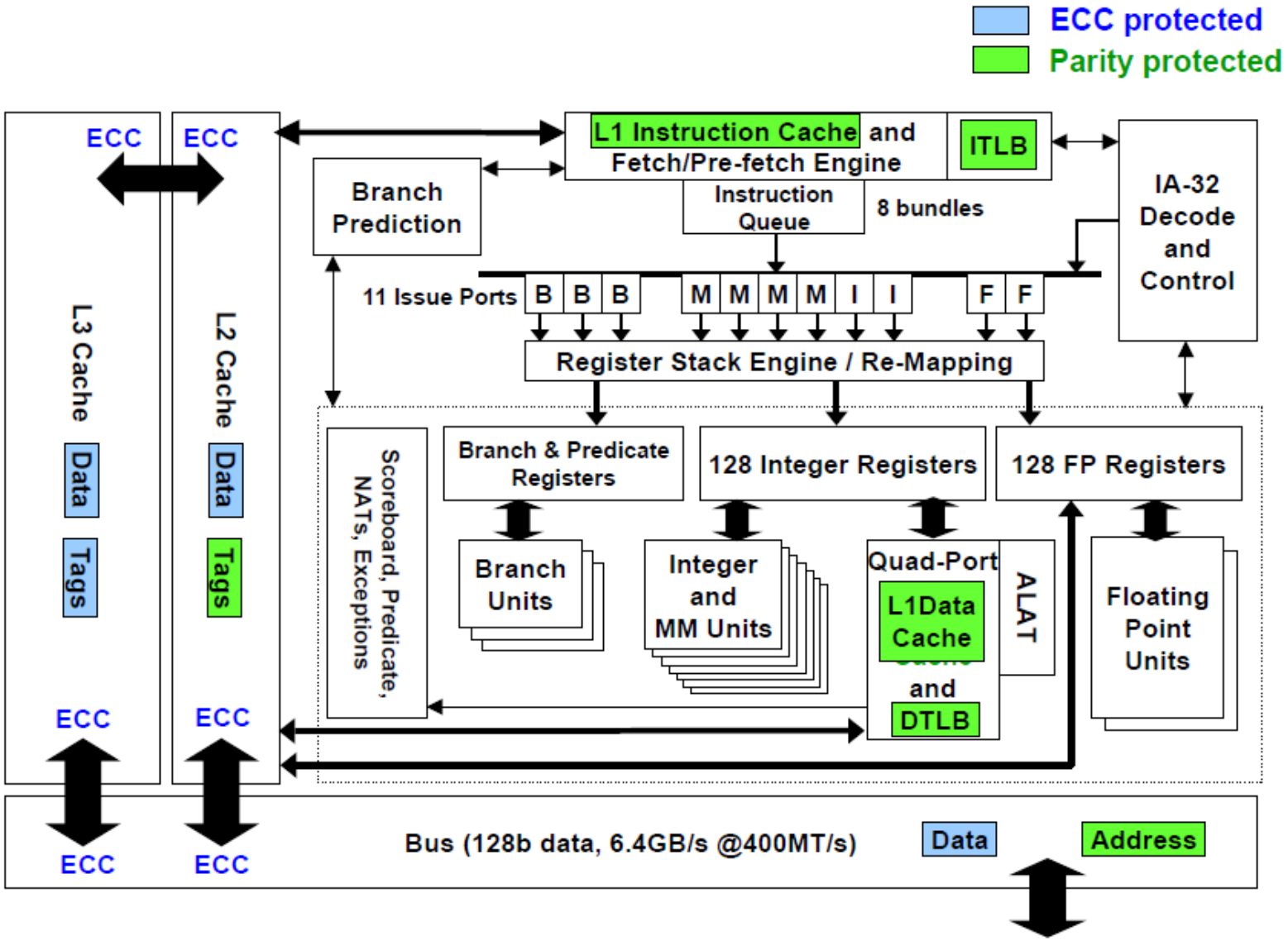
The Madison 6 MB processor of the Itanium 2 platform is a scaled down versions (from 180 nm to 130 nm) of the McKinley processor.

This results in

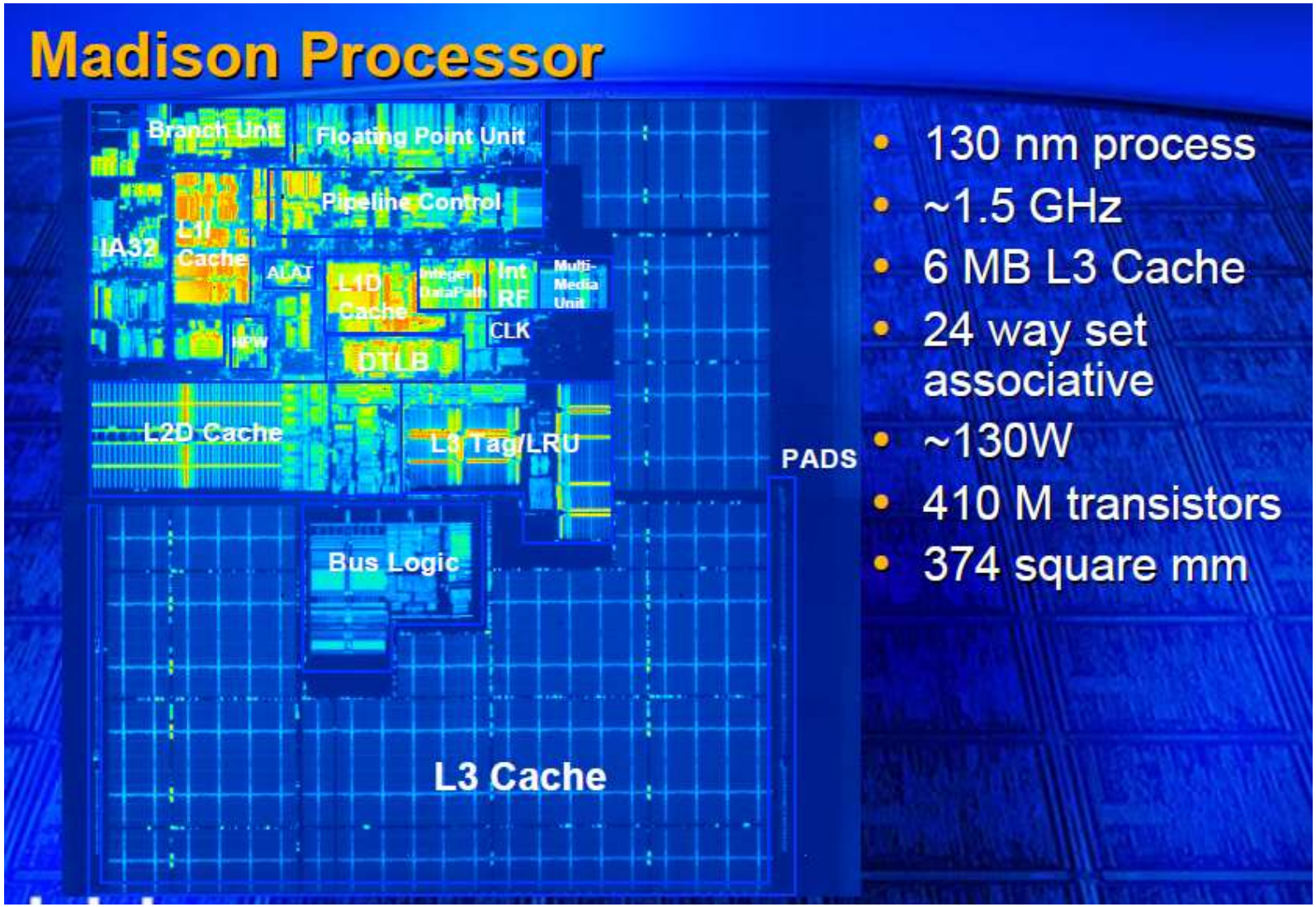
- higher clock speed (up to 1.5 GHz vs. 1.0 or 0.9 GHz)
- larger L3 cache size (up to 6 MB vs. 3 or 1.5 MB) L3.

5.3 The Itanium 2 Madison 6 MB L3 line (3)

Block diagram of the Madison 6M with indicating data protection features [66

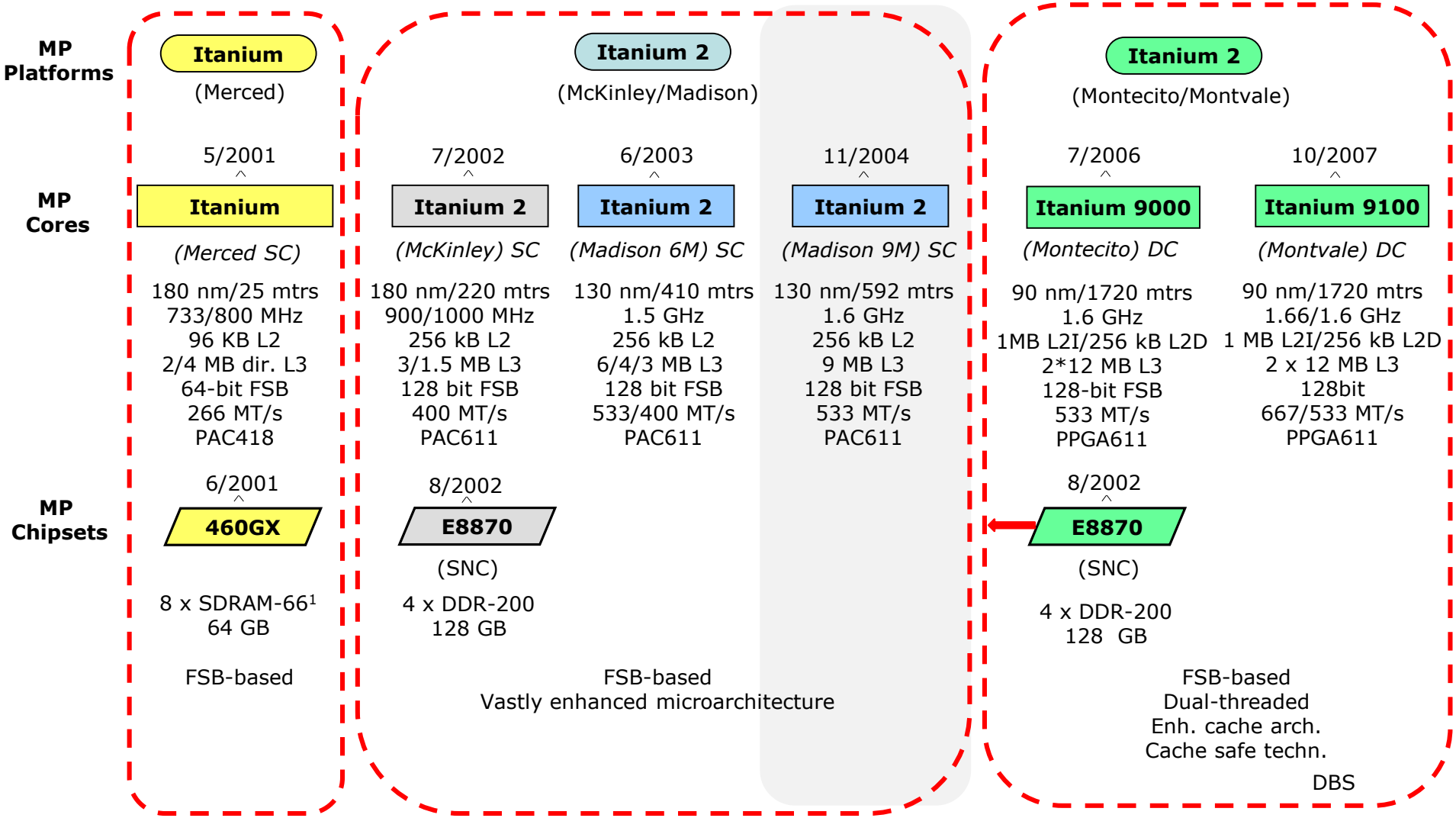


Die plot of the Itanium 2 Madison 6 MB processor [33]



5.4 The Itanium 2 Madison line with 9 MB L3

5.4 The Itanium 2 Madison 9M line-1



[1]: Special memory cards are used SNC: Scalable Node Controller

5.4 The Itanium 2 Madison line with 9 MB L3 (2)

The Itanium 2 Madison 9 M line

Introduction: 11/2004
Introduced models

- 1.6 GHz 9 MB L3
- 1.6 GHz 6 MB L3
- 1.6 GHz 3 MB L3
- 1.5 GHz 4 MB L3

The Itanium 2 Madison 9 MB line of processors are 130 nm parts with an improved process technology, resulting in

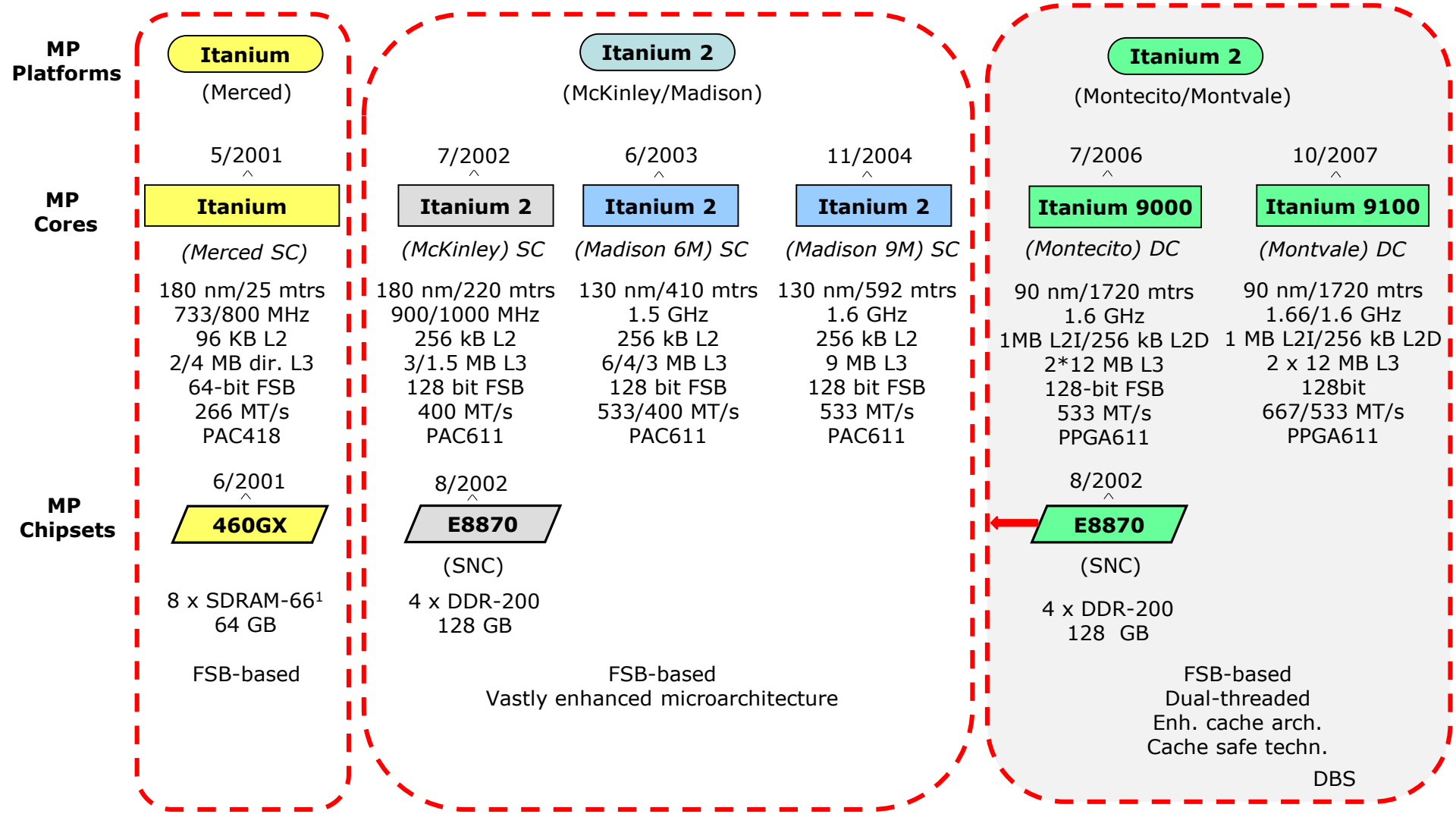
- higher FSB speed (533 MT/s vs. of 400 MT/s) and
- larger L3 cache size (9 MB vs 6 MB).

6. The Itanium 2 Montecito/Montvale DC platform

- 6.1 Overview of the Itanium 2 Montecito/Montvale DC platform
- 6.2 The Itanium 9000 (Montecito) line
- 6.3 The Itanium 9100 (Montvale) line

6.1 Overview of the Itanium 2 Montecito/Montvale DC platform

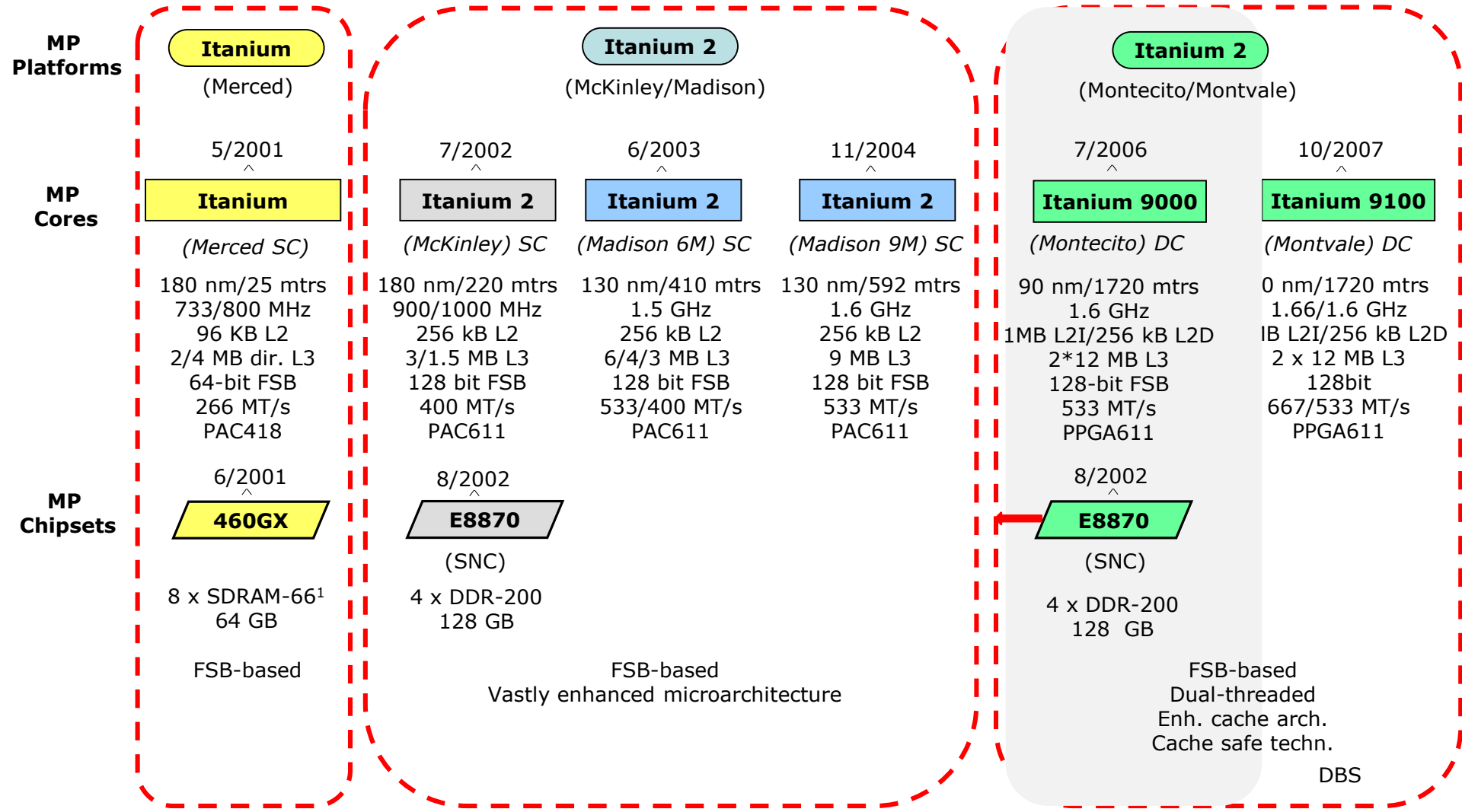
6.1 Overview of the Itanium 2 Montecito/Montvale platform



[1]: Special memory cards are used SNC: Scalable Node Controller

6.2 The Itanium 9000 (Montecito) line

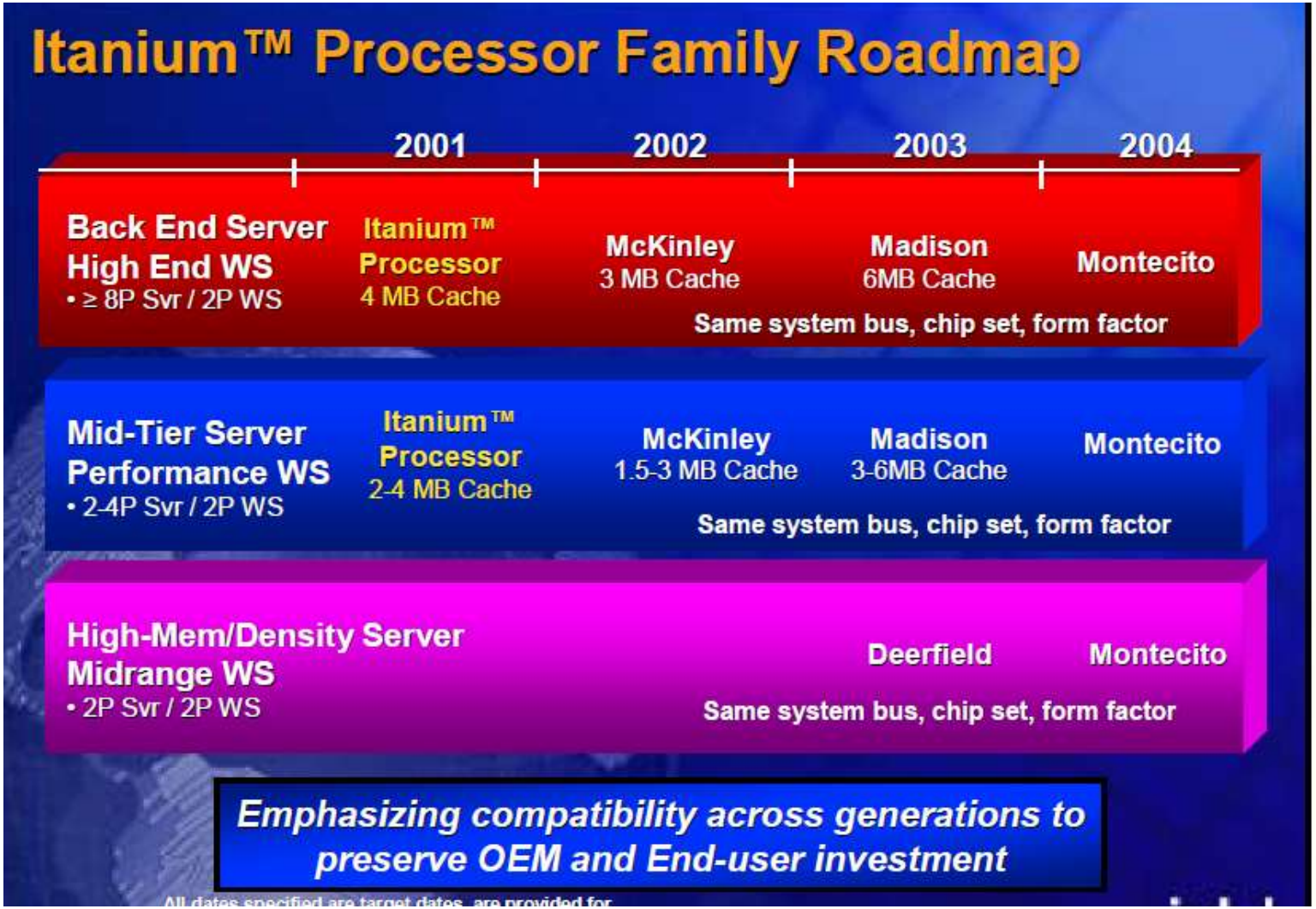
6.2 The Itanium 9000 (Montecito) line



[1]: Special memory cards are used SNC: Scalable Node Controller

The Itanium 9000 (Montecito) line-2

At introducing the Itanium line (2001) it was scheduled for 2004, as seen below [11].



The Itanium 9000 (Montecito) line-3

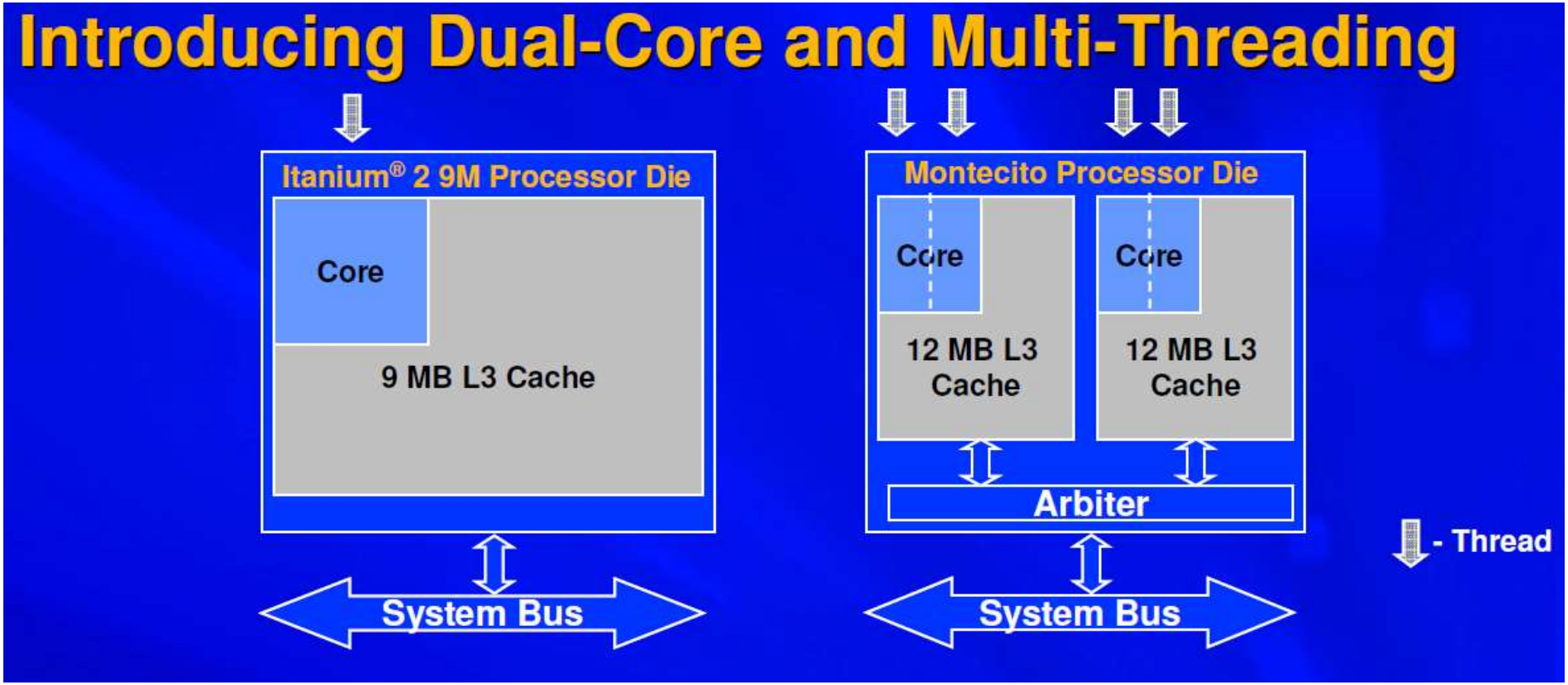
- Introduced: 7/2006 with two years delay to the original schedule.
- Manufactured with 90 nm feature size, 1720 mtrs
- Drop-in replacement for previous Itanium 2 Madison processors.

Key enhancements

a) dual-core implementation
b) 2-way multithreading (called Temporal Multi-Threading)
c) Enhanced cache architecture
d) Cache safe technology (for the L3 cache)
e) Vernier technology to reduce clock skews
f) Hardware support for virtualization (not discussed)

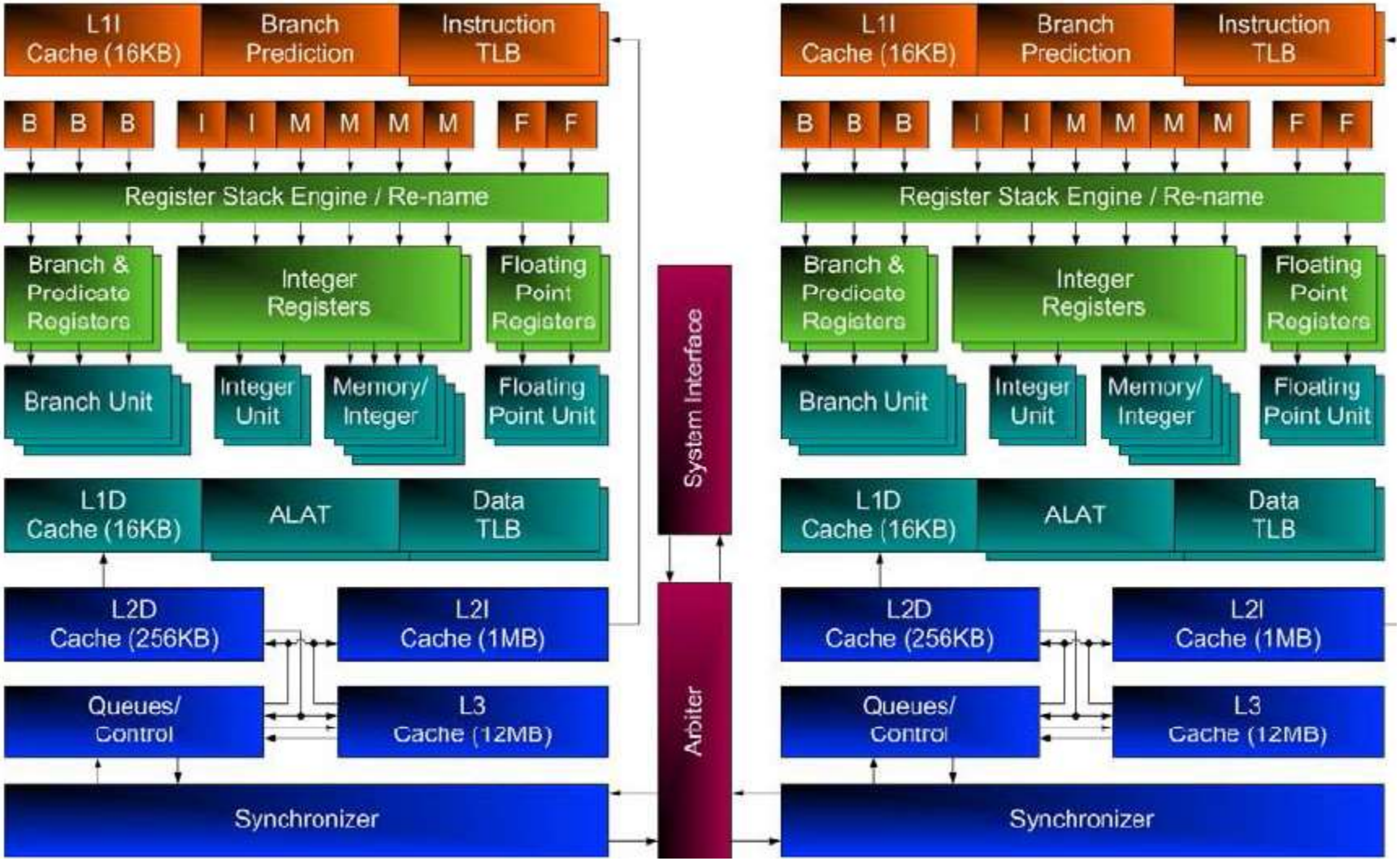
a) Dual core implementation

High-level block diagram of Montecito [36]



6.2 The Itanium 9000 (Montecito) line (5)

Block diagram of the Itanium-9000 (Montecito) processor [37]



b) 2-way coarse grain multithreading (called Temporal Multi-Threading) [38]

Principle of thread switching

A thread switch will be initiated typically when a thread stall event occurs, as indicated below.

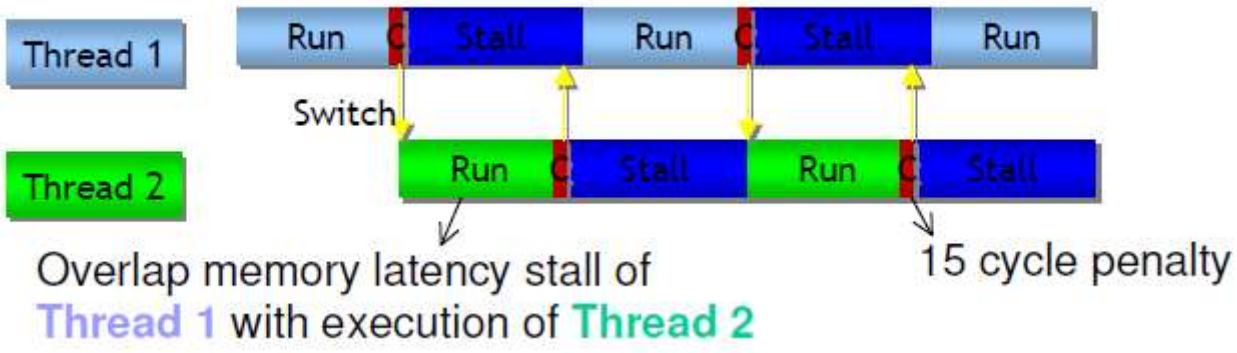
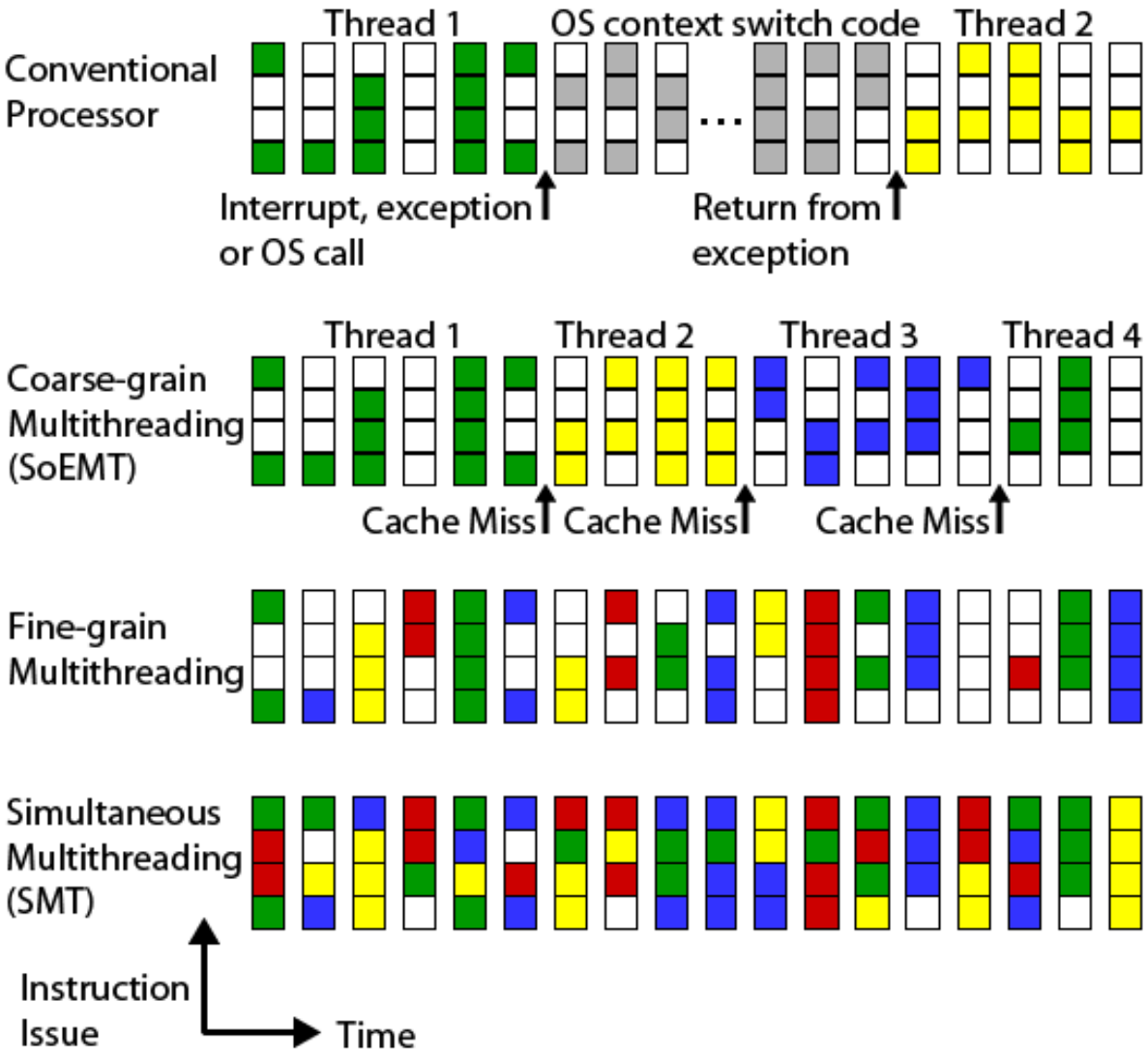


Figure: Principle of thread switching in Montecito [38]

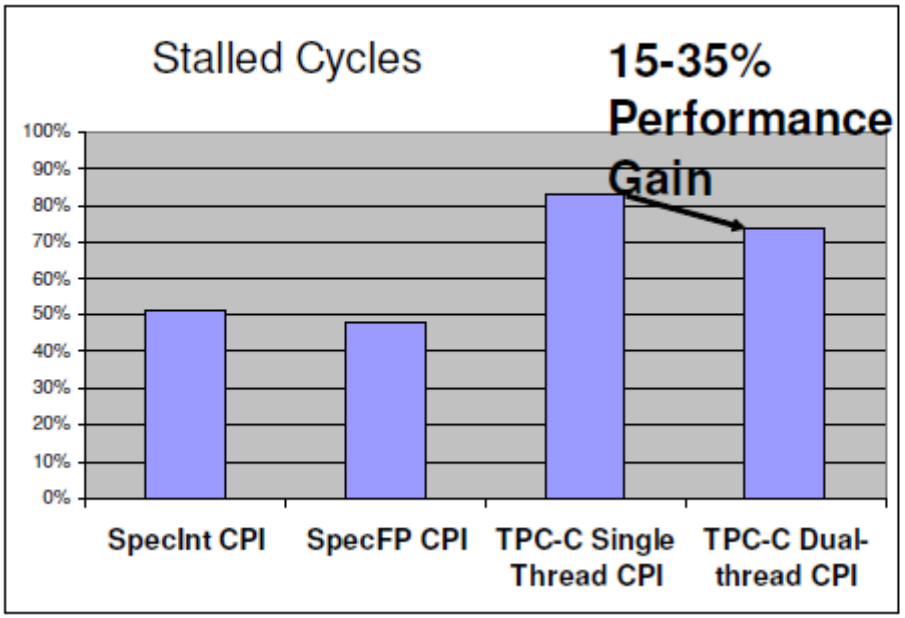
Multithreading strategies [31]



SoEMT: Switch-on-event multi-threading (used in Montecito and subsequent Itaniums)

Events: L3 cache misses etc.

The rate of stalled cycles and the achievable gain [38]



2% core area impact and 0% cycle time impact

Core area impact: ~ 2 %

6.2 The Itanium 9000 (Montecito) line (9)

Thread switch events [36]

- L3 miss/return
  (instruction, data or hypervisor access)
- Time slice expiration
- Low power state entered/excited
- Switch hint execution
- ALAT invalidation

Microarchitecture support for multithreading in Montecito

- We note that the microarchitecture of Montecito supports 2-way multithreading by duplicating the architectural and microarchitectural states, as indicated in the block diagram of Montecito above.
- The architectural state is duplicated by dual integer, FP, predicate and branch registers, whereas

  the microarchitectural state is duplicated by dual return stack buffer (not shown), dual advanced load address table (ALAT) and performance monitoring.
- On the other hand both threads share the execution resources and the memory hierarchy (caches and TLBs (Translation Look Aside Buffers)).

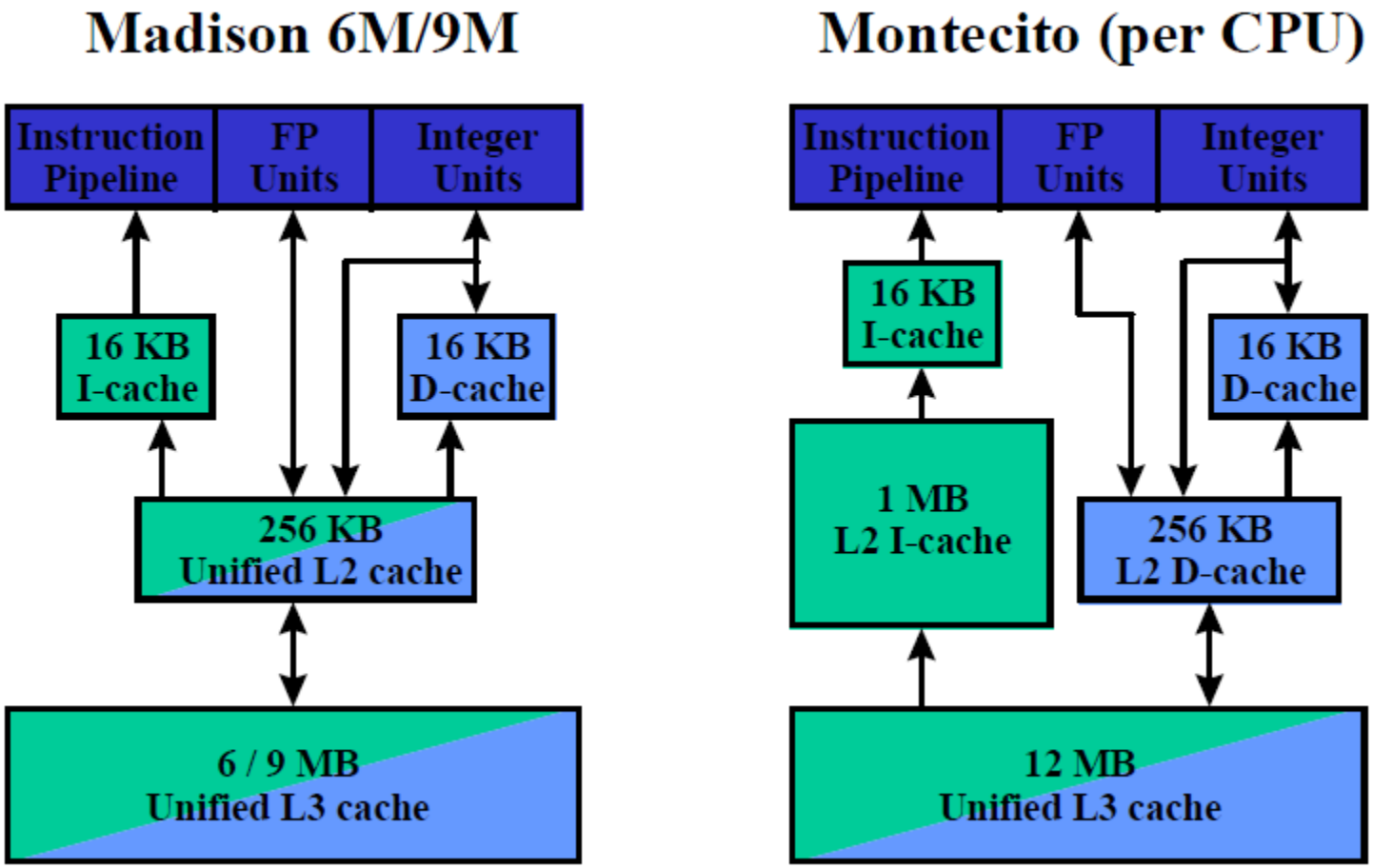
c) Enhanced cache hierarchy-1 [67]

The Montecito has

- split L2 caches (1 MB L2I and 256 kB L2D) per core
  rather than a shared L2 cache both for instructions and data as for preceding Itanium implementations
- up to 12 MB/core L3 cache

as shown in the next Figure.

c) Enhanced cache hierarchy-2 [67]

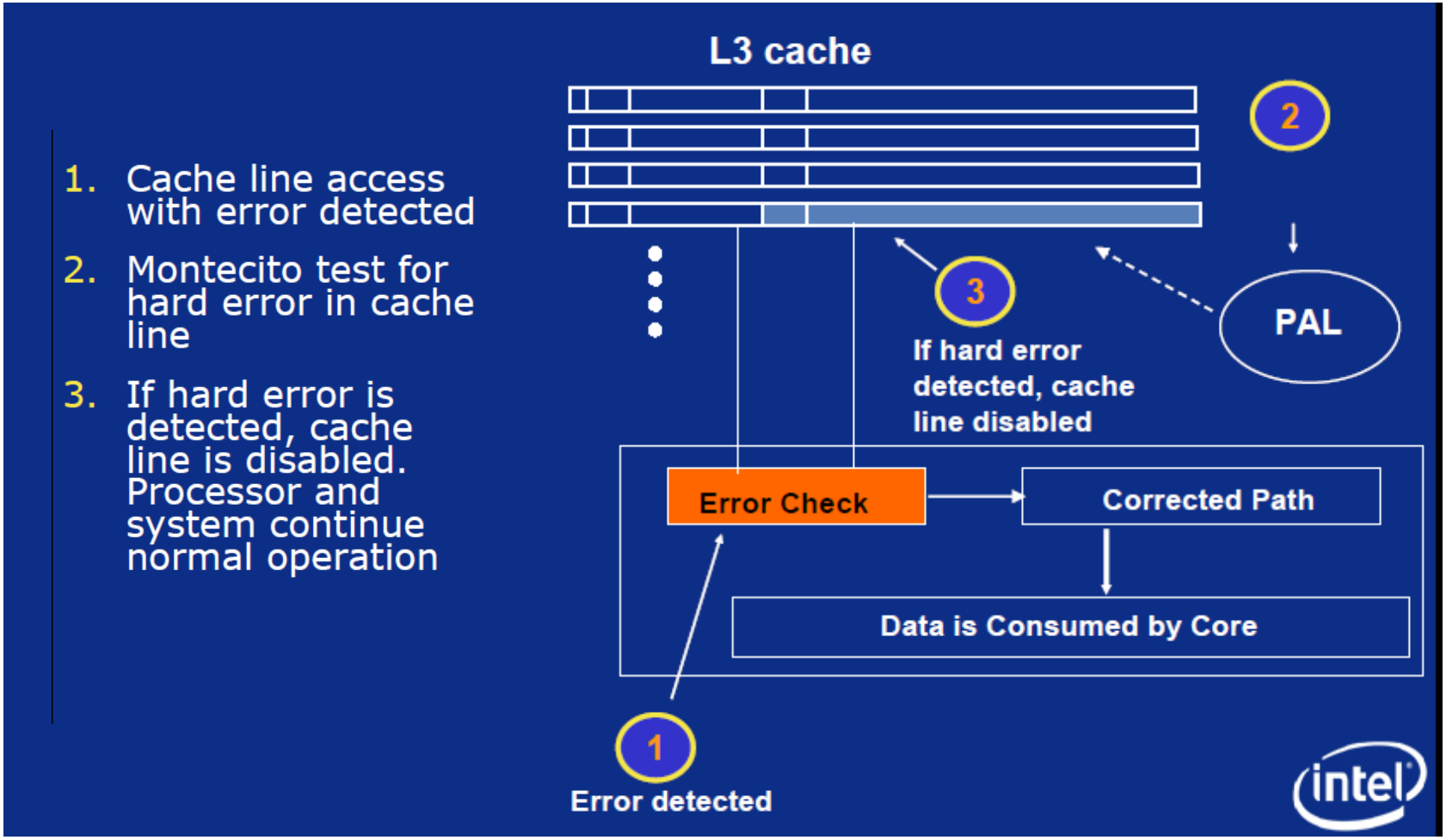


d) The cache safe technology

Previously designated as the Pellston technology.

In Montecito Intel introduced the Cache safe technology for the L3 cache and in their 4-core Tukwila extended it also to both the L2I, L2D and the Directory cache (Snoop Filter).

Principle of operation of the Cache Safe technology [36]

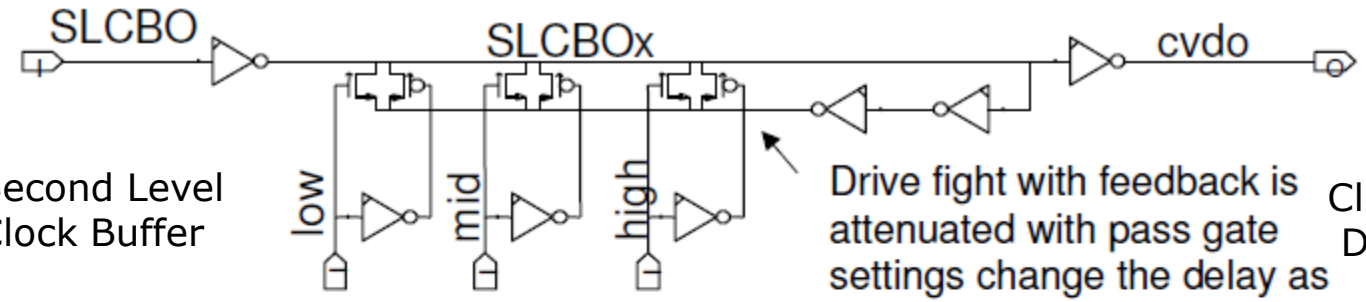


6.2 The Itanium 9000 (Montecito) line (15)

e) Vernier technology to reduce clock skews [39]

- Traditional clock trees use simple buffers to distribute clock signals.
- Verniers are digitally programmable buffers that allow delay insertion as needed.
- Montecito make use of 7536 verniers per core.
- Verniers allow 70ps of adjustment (about 1/10 of the clock period).
- The adjustments are made after production and help to compensate variations on the die.

Clock vernier circuit an resulting delayed signal forms [39]

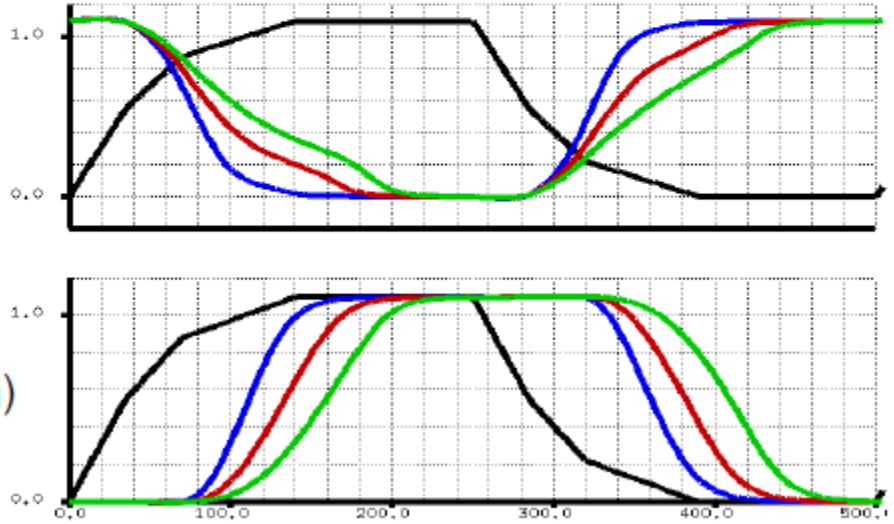


Second Level Clock Buffer

Drive fight with feedback is attenuated with pass gate settings change the delay as desired

Clock Vernier Device

SPICE simulations showing low, mid and high delay settings for SLCBOx (top graph) and CVDO (bottom graph)



Itanium 9000 (Montecito) models introduced at launch [40]

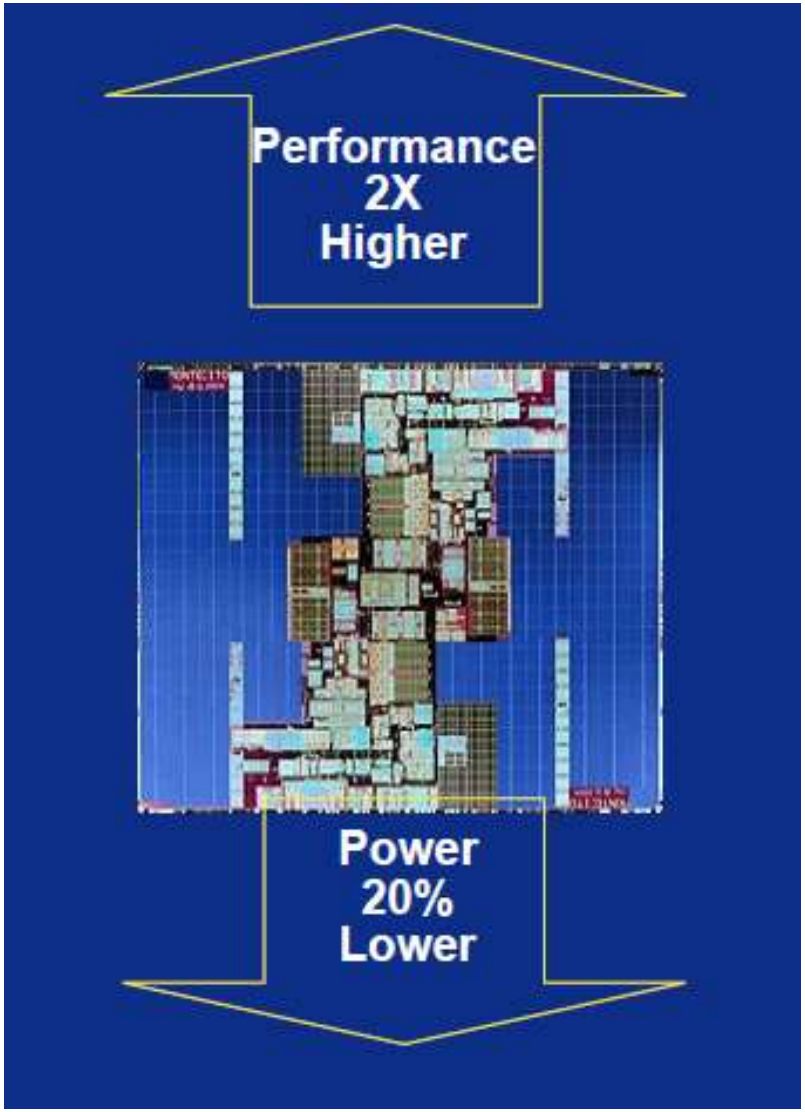
Intel Itanium 2 "Montecito" Processors					
Processor Number	Core Frequency	Bus Frequency	L3 Cache	TDP	Price
9050	2 x 1.6 GHz	533 MHz	24MB	104W	$3,692
9040	2 x 1.6 GHz	533 MHz	18MB	104W	$1,980
9030	2 x 1.6 GHz	533 MHz	8MB	104W	$1,552
9020	2 x 1.42 GHz	533 MHz	12MB	104W	$910
9015	2 x 1.4 GHz	400 MHz	12MB	104W	$749
9010	1.6 GHz	533 MHz	6MB	75W	$696

6.2 The Itanium 9000 (Montecito) line (18)

The operating frequency (FREQ) – core voltage (VCORE) diagram (Shmoo diagram) for the Montecito processor [38]

Power Consumption is for all 4 voltage domains at ~40C Tj

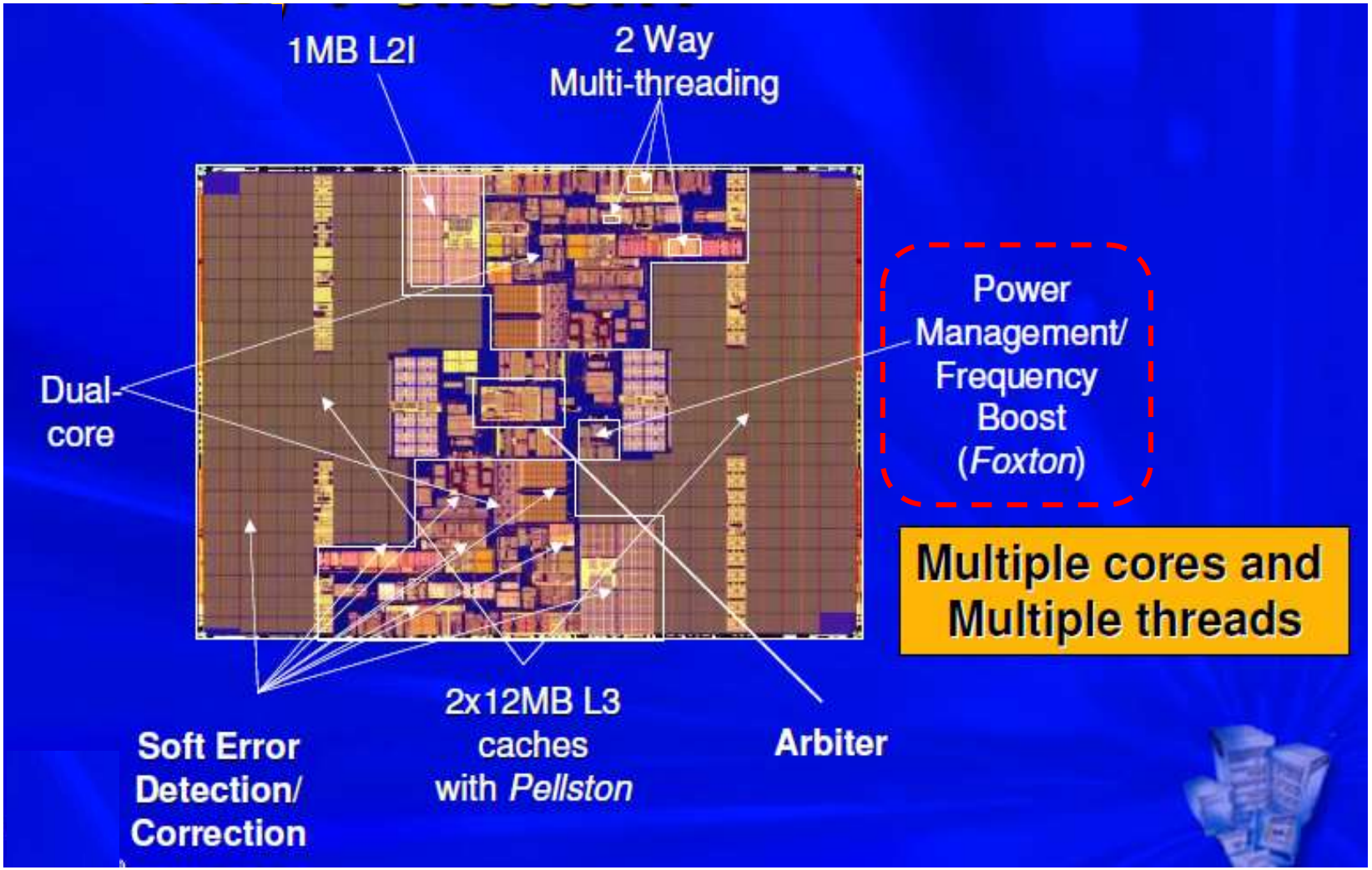
Power/performance expectations for Montecito [41]



Planned introduction and final cancellation of the Foxton technology

The Foxton technology was planned to be introduced into the Montecito processor but Intel withdraw it due to implementation problems.

Intel's intention to introduce the Foxton technology [36]



Motivation for introducing the Foxton technology-1 [42]

Different applications generate different amount of power, as the next Figure indicates.

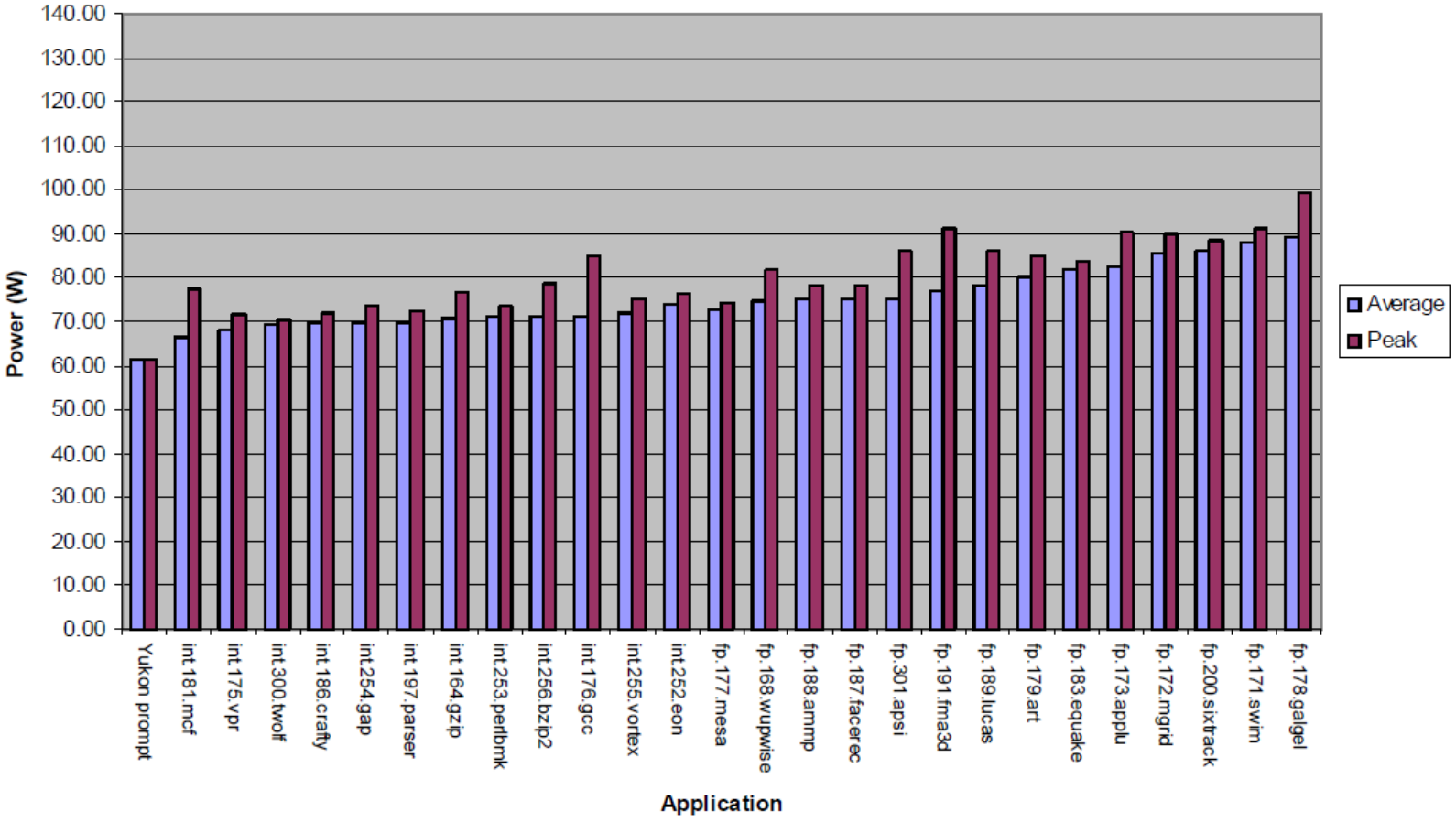


Figure: Total power for various applications [42]

Motivation for introducing the Foxton technology-2 [42]

As the Figure shows, when the operating voltage and frequency would be set on a worst case principle, i.e. according to the most demanding application (*fp.178.galgel*), then "low activity workloads" would not fully utilize the available power budget and would run at a lower frequency than would be possible when utilizing the entire power budget.

Foxton aims at utilizing the full power budget also for "light workloads" by increasing the clock frequency and thus performance in these cases by about 10 – 15 %.

Principle of the operation of the Foxton technology [42]

Foxton consists of two control subsystems,

- The power and temperature subsystem and
- the frequency subsystem,

as indicated below.

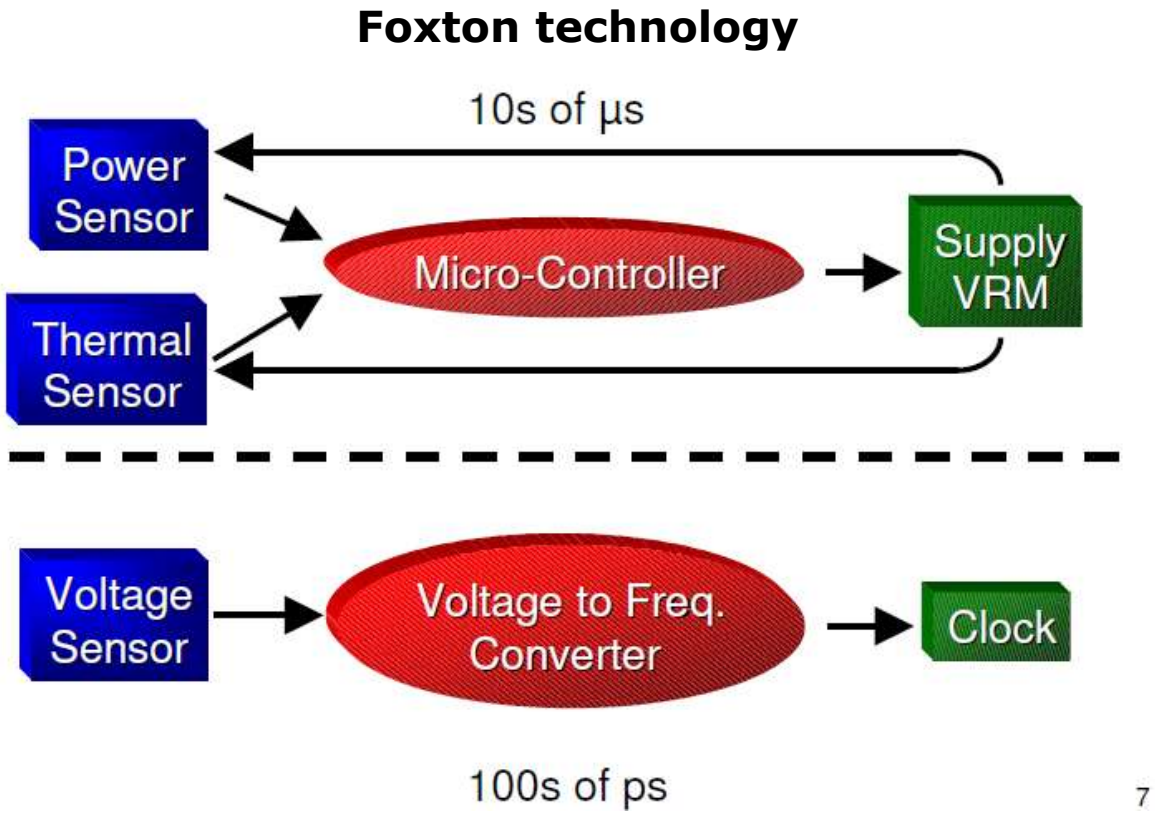


Figure: high level operation of the Foxton technology [42]

The power and temperature subsystem-1 [42]

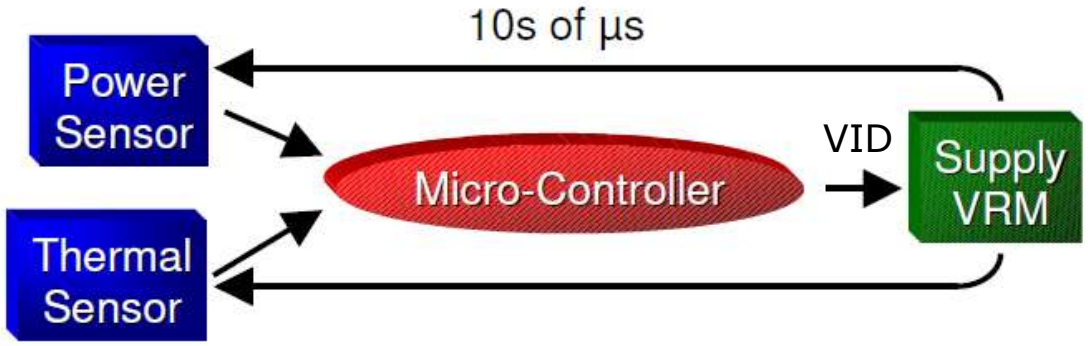


Figure: High level view of the power and temperature subsystem [42]

- *This subsystem* monitors both power and temperature.
- Power and temperature inputs are fed into an embedded 16-bit micro-controller that detects if there is an over-power or over-temperature event.

  In these cases it reduces the core voltage by changing the VID (Voltage ID) bits of the power supply VRM (Voltage Regulation Module).
- The sensors then continue to monitor power and temperature and, as the over power/temp event subsides, the micro-controller raises the voltage back to the minimum necessary to maintain the part's maximum frequency.

  There are 6 VID bits to provide a full dynamic range between 0.8V and 1.3V in 12.5 mV increments for both the core and the cache.

The power and temperature subsystem-2 [42]

- To the opposite, when the power and temperature values indicate a headroom, the microcontroller may raise the core voltage to give the chance to increase the clock frequency and thus performance.
- The power and frequency subsystem operates on the order of 100s of μs.
- Foxton modifies only the core voltage.
- The cache is set to the minimum voltage needed for cache stability to minimize leakage.
- The IO supply voltage remains constant.

The frequency control subsystem [42]

- It continuously monitors core voltage to see if the current voltage is appropriate for the current core frequency.

  If the voltage is too low for the current frequency, the subsystem lowers the clock frequency to a value that the voltage will support.

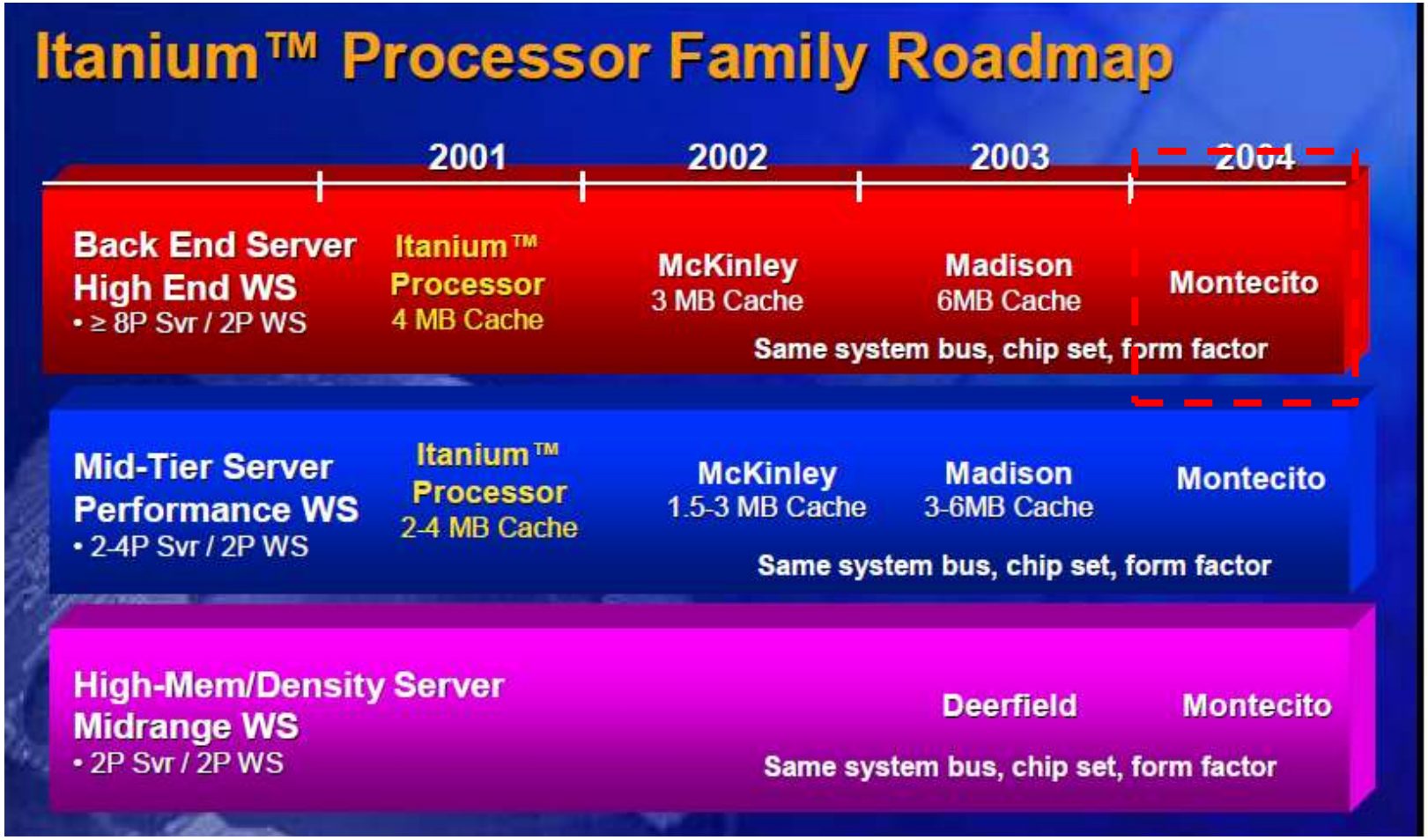
  This happens in 1 or two clock cycles.
- When the core voltage is high enough, the frequency will again be raised to maximize performance.
- We note that both subsystems are connected only indirectly through the core voltage.

  The first subsystem sets the core voltage according to the measured power and temperature values whereas the second subsystem sets the core frequency according to the set core voltage.

Utilization of the Foxton technology

- Despite numerous announcements about introducing the Foxton technology into the Montecito processor at various events (including ISSCC 2005 or Hot Chips 2005), Intel did not utilize this technology neither in their Montecito or the subsequent Montvale processor.
- By contrast, the next member of the Itanium family, the Tukwila (Itanium 9300) introduced a competing adaptive technology for raising the clock frequency when there is a power headroom (the Turbo Boost technology).
- Here we note that according to various sources (e.g. [43]) the sophistication of the Foxton technology may be claimed for the about two years delay in launching the Montecito processor, since in 2002 Intel planned its introduction yet for 2004 (as the subsequent Figure shows) but its launch became delayed to 2006.

Intel's Itanium roadmap from 4/2002 [11]



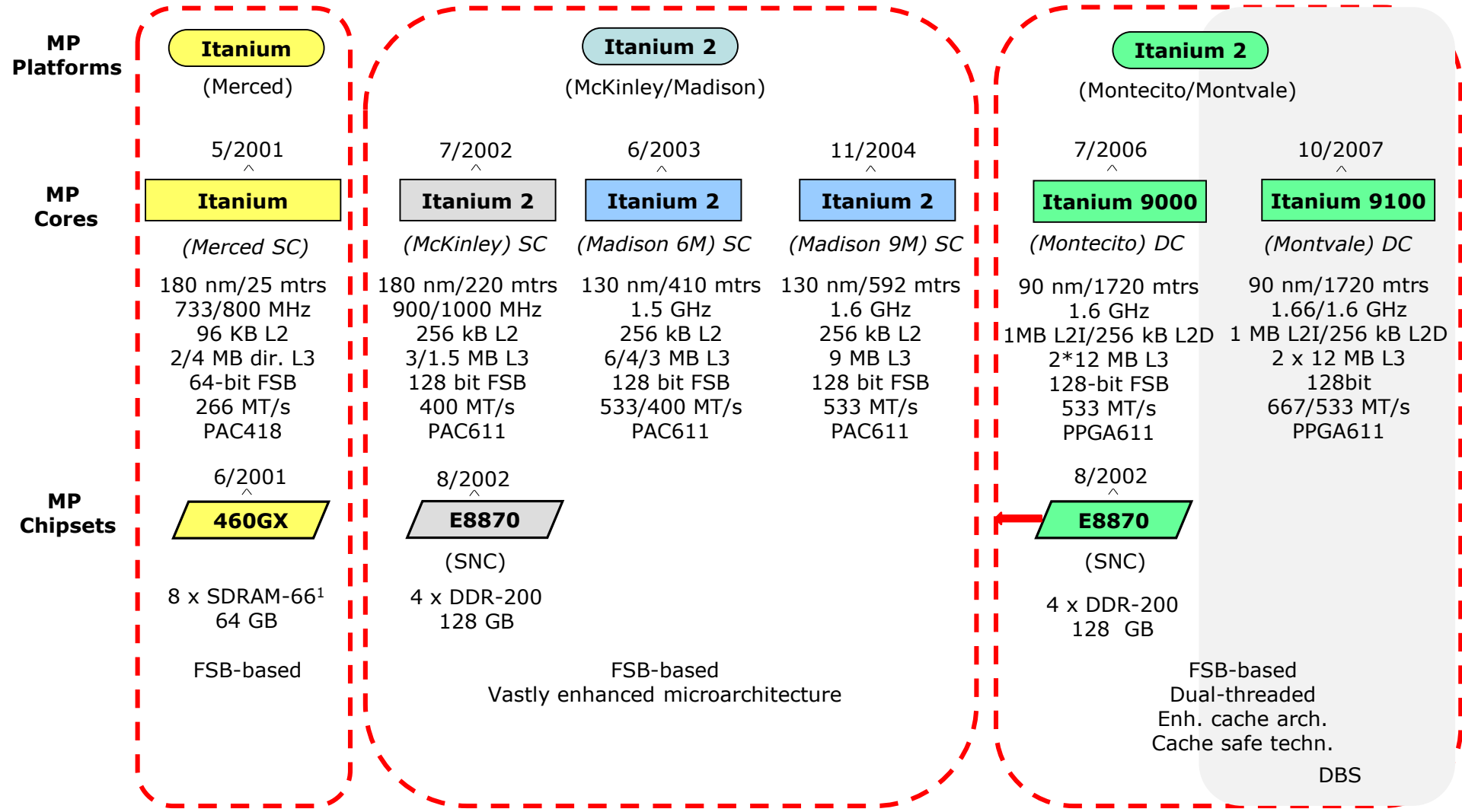
6.2 The Itanium 9000 (Montecito) line (30)

Remark on the departure of the director of Itanium circuits and technology (Sam Naffziger) to AMD [44]

- Sam Naffziger led the Itanium design team of HP for eight years.
- In 2005 he joined Intel as part of a group of Itanium designers.
- About 3/2006 he and 7 other Itanium designers were hired away by AMD.

6.3 The Itanium 9100 (Montvale) line

6.3 The Itanium 9100 (Montvale) line



[1]: Special memory cards are used SNC: Scalable Node Controller

6.3 The Itanium 9100 (Montvale) line (2)

The Itanium 9100 (Montvale) line

- Introduced: 10/2007
- Manufactured with 90 nm feature size
- Drop-in replacement for previous Montecito processors.

Main enhancements

- The Montvale processor line provides only minor improvements vs. its predecessor (Montecito).
- It has an about 10 % higher maximum clock speed (1.66 GHz vs. 1.6 GHz) and about 25 % higher FSB speed (667 MT/s vs. 533 MT/s).
- Further on, the Montvale line introduced the DBS (Demand Based Switching) technology to reduce power consumption.

  We note that DBS dynamically modifies the core voltage and clock frequency such that the actual workload should be performed by the lowest possible clock frequency and thus power consumption.

  DBS is the server equivalent of Intel's Enhanced SpeedStep technology (EIST) introduced earlier for mobiles and desktops.

6.3 The Itanium 9100 (Montvale) line (3)

Itanium 9100 (Montvale) models introduced at launch [45]

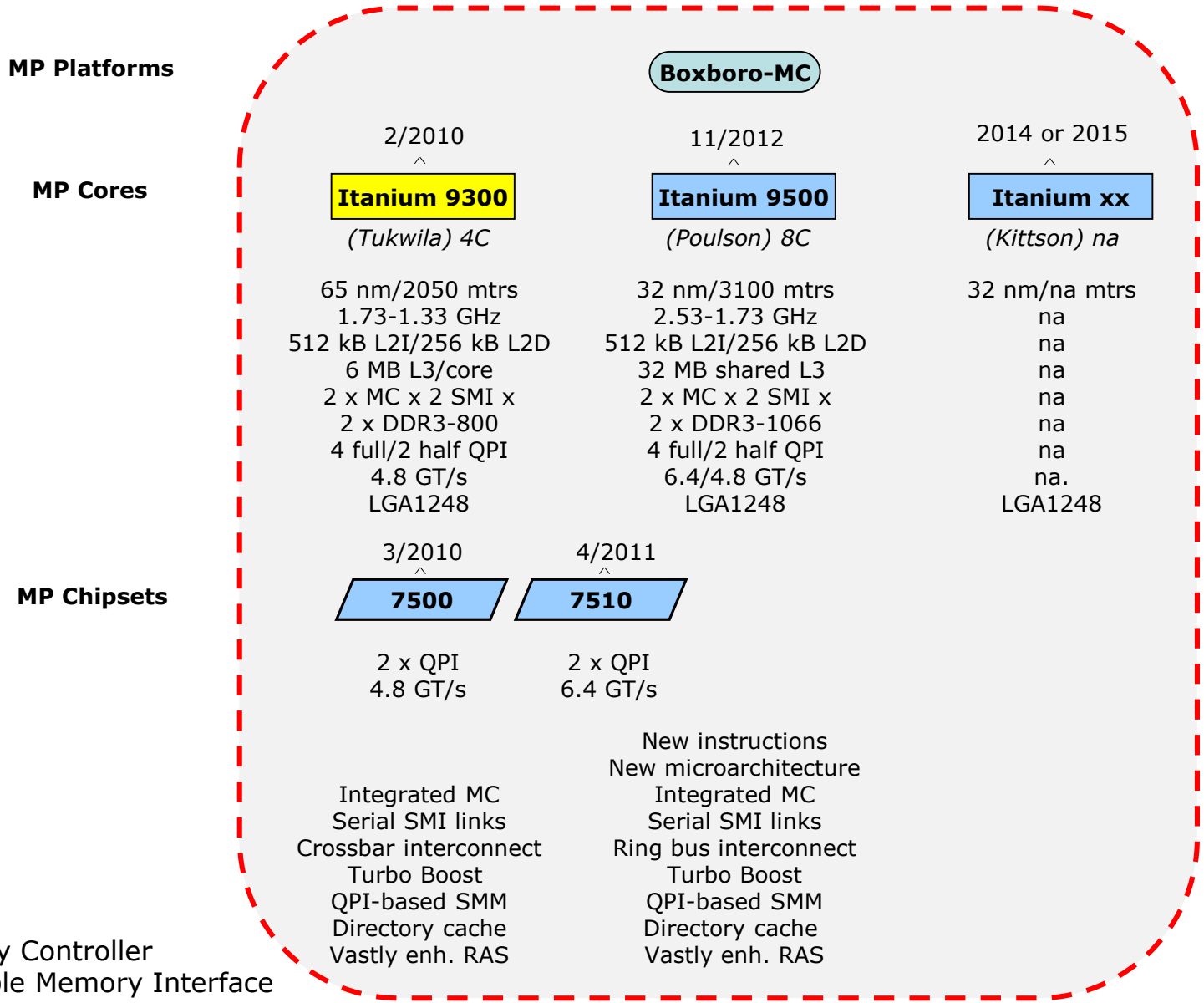
Intel Dual-Core Itanium 2				
Processor Number	Clock Speed	FSB	iL3 Cache	Q4'07 Launch Price
9150M	1.66	667 MHz	24MB	$3692
9150N	1.60	533/400 MHz	24MB	$3692
9140M	1.66	667 MHz	18MB	$1980
9140N	1.60	533/400 MHz	18MB	$1980
9130M	1.66	667 MHz	8MB	$1552
9120N	1.42	533/400 MHz	12MB	$910

7. The Boxboro MC platform (4-core Tukwila/8-core Poulson/Kittson)

- 7.1 Overview of the Boxboro MC platform (4-core Tukwila/8-core Poulson/Kittson)
- 7.2 The 4-core Tukwila (Itanium 9300) line
- 7.3 The 8-core Poulson (Itanium 9500) line
- 7.4 The Kittson line

7.1 Overview of the Boxboro MC platform (4-core Tukwila/8-core Poulson/Kittson)

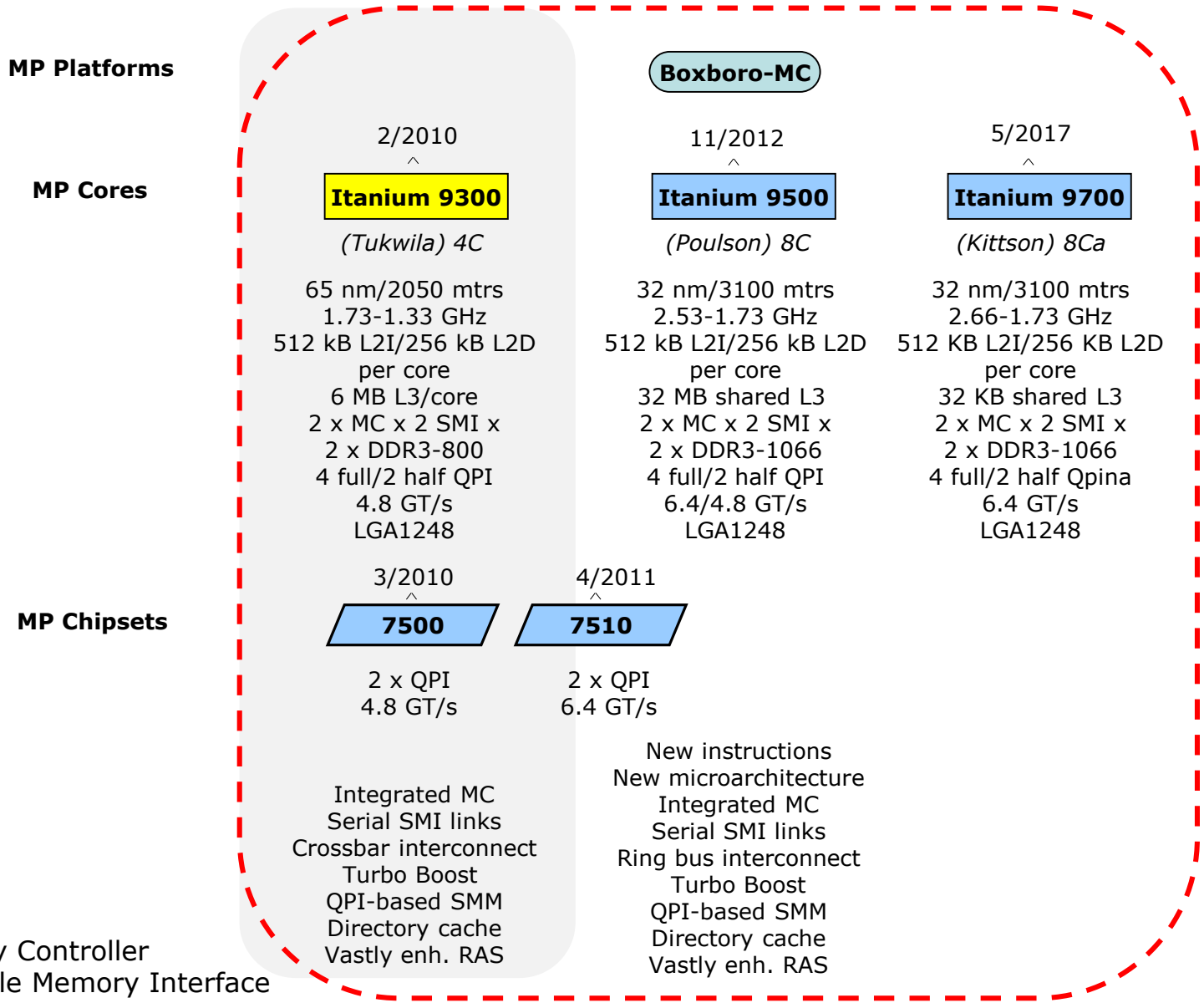
7.1 Overview of the Boxboro-MC platform



MC: Memory Controller
SMI: Scalable Memory Interface

7.2 The 4-core Tukwila (Itanium 9300) line

7.2 The 4-core Tukwila (Itanium 9300) line-1

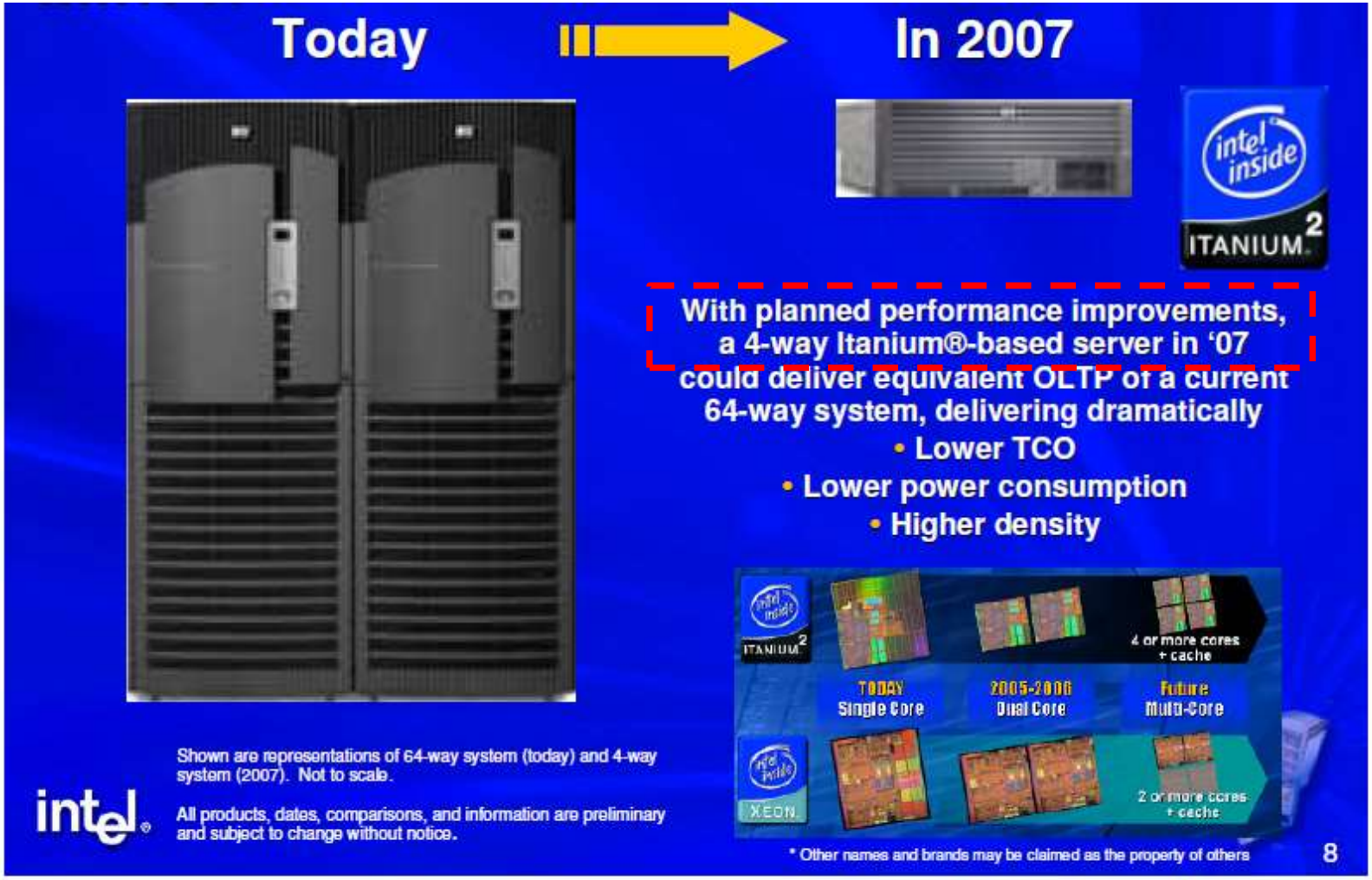


MC: Memory Controller
SMI: Scalable Memory Interface

The 4-core Tukwila (Itanium 9300) line-2

- Planned introduction date (in 2005): 2007 (as indicated in the next Figure).
- Date of shipping: 2/2010

Planned introduction date of the Tukwila processor (in 2005) [36]



7.2 The 4-core Tukwila (Itanium 9300) line (4)

The Itanium 9300 (Tukwila)

- Manufactured with 65 nm feature size, 2050 mtrs
- New socket (LGA1248)

Key enhancements

a) 4 cores

b) Integrated memory controllers

c) Serial memory connection through SMI links
2 MC x2 SMI link x2 DDR-800

d) Turbo Boost technology

e) QPI-based SMP interconnect
(4x full width + 2x half-width/4.8 Gb/s)

f) On-die directory based cache coherency
(2x 1 MB on-die directories (snoop filters)

g) Vastly enhanced RAS features

h) Multi-threading enhancements

Performance

Up to 2x performance of the Itanium 9000 series

7.2 The 4-core Tukwila (Itanium 9300) line (5)

Remark

A number of key enhancements of Tukwila, such as

- 4 cores
- Integrated memory controllers
- Turbo Boost technology and
- QPI-based SMP interconnect

were previously introduced by Intel in their 1. generation Nehalem (Core i7-9xx (Bloomfield)) desktop line (11/2008), whereas

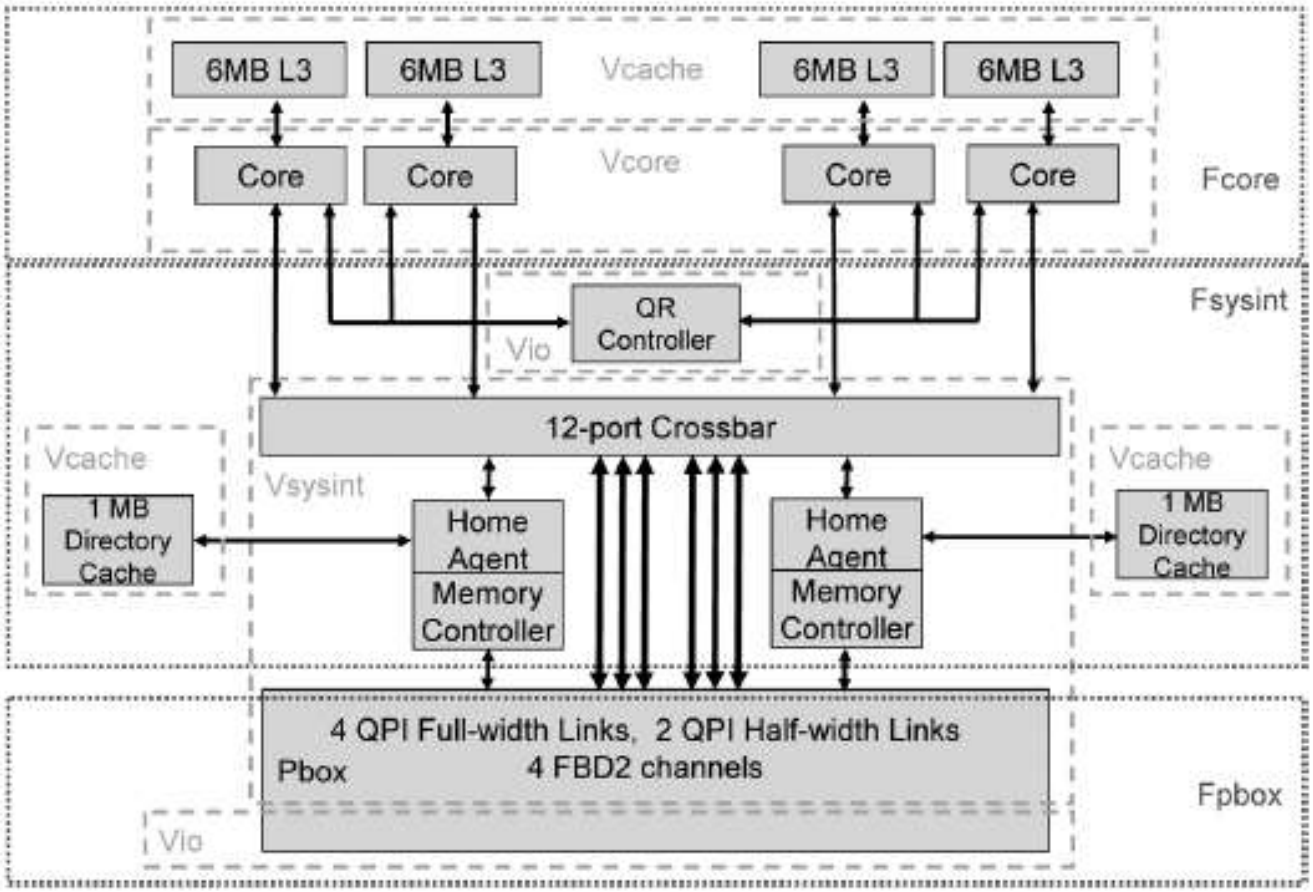
- Serial memory connection through SMI links

debuted by Intel in their Nehalem-EX (Xeon 7500 (Beckton)) server line (3/2010).

7.2 The 4-core Tukwila (Itanium 9300) line (6)

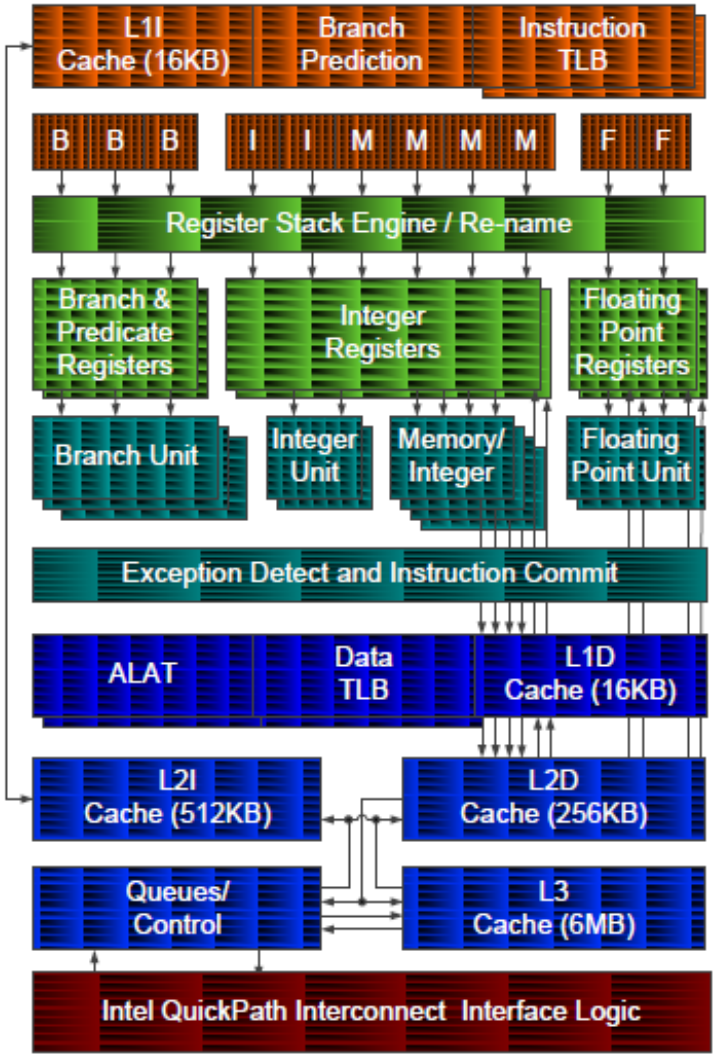
a) 4 Cores

Block diagram of the Itanium 9300 (Tukwila) processor (2010) [46]

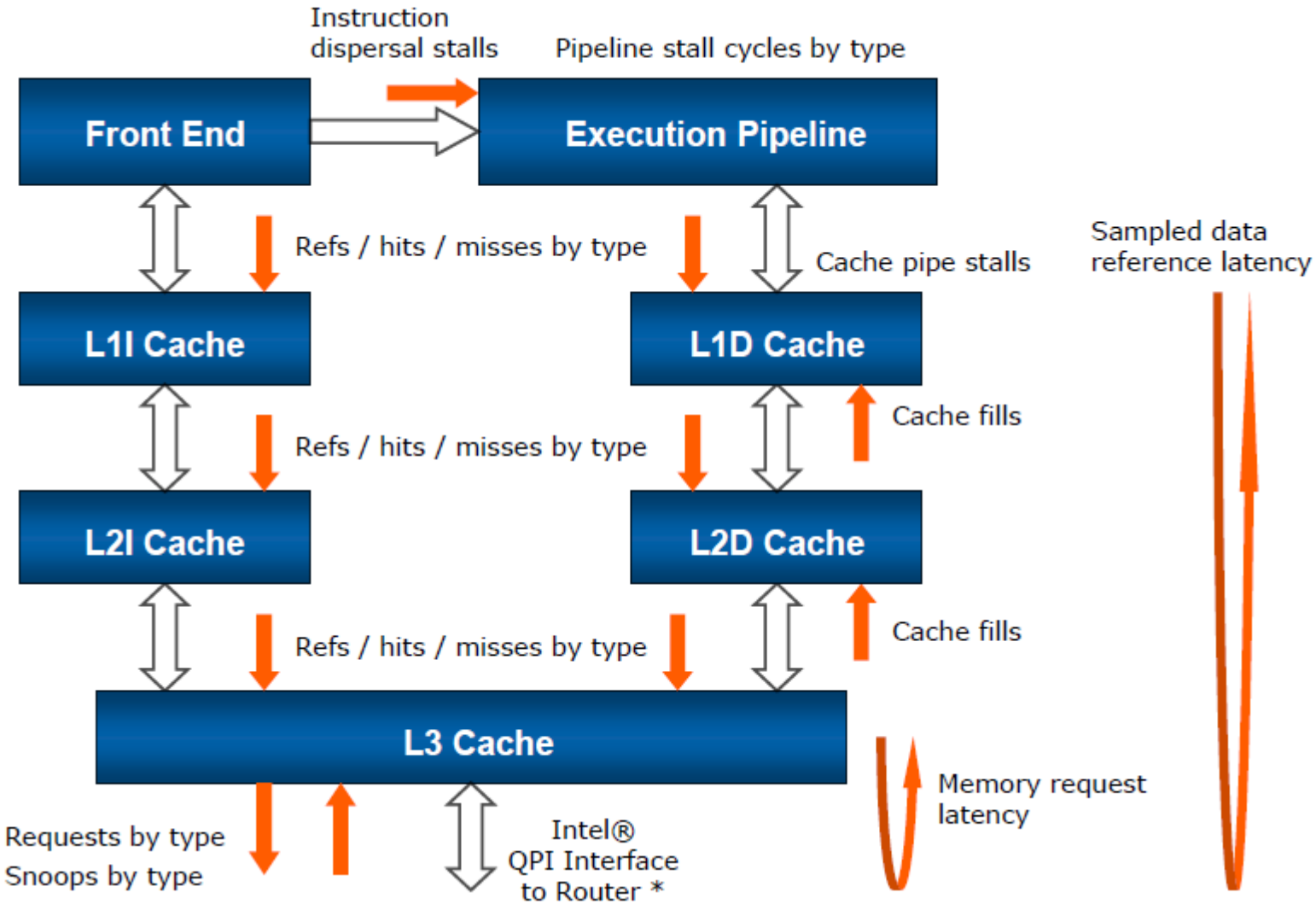


Block diagram of the cores [47]

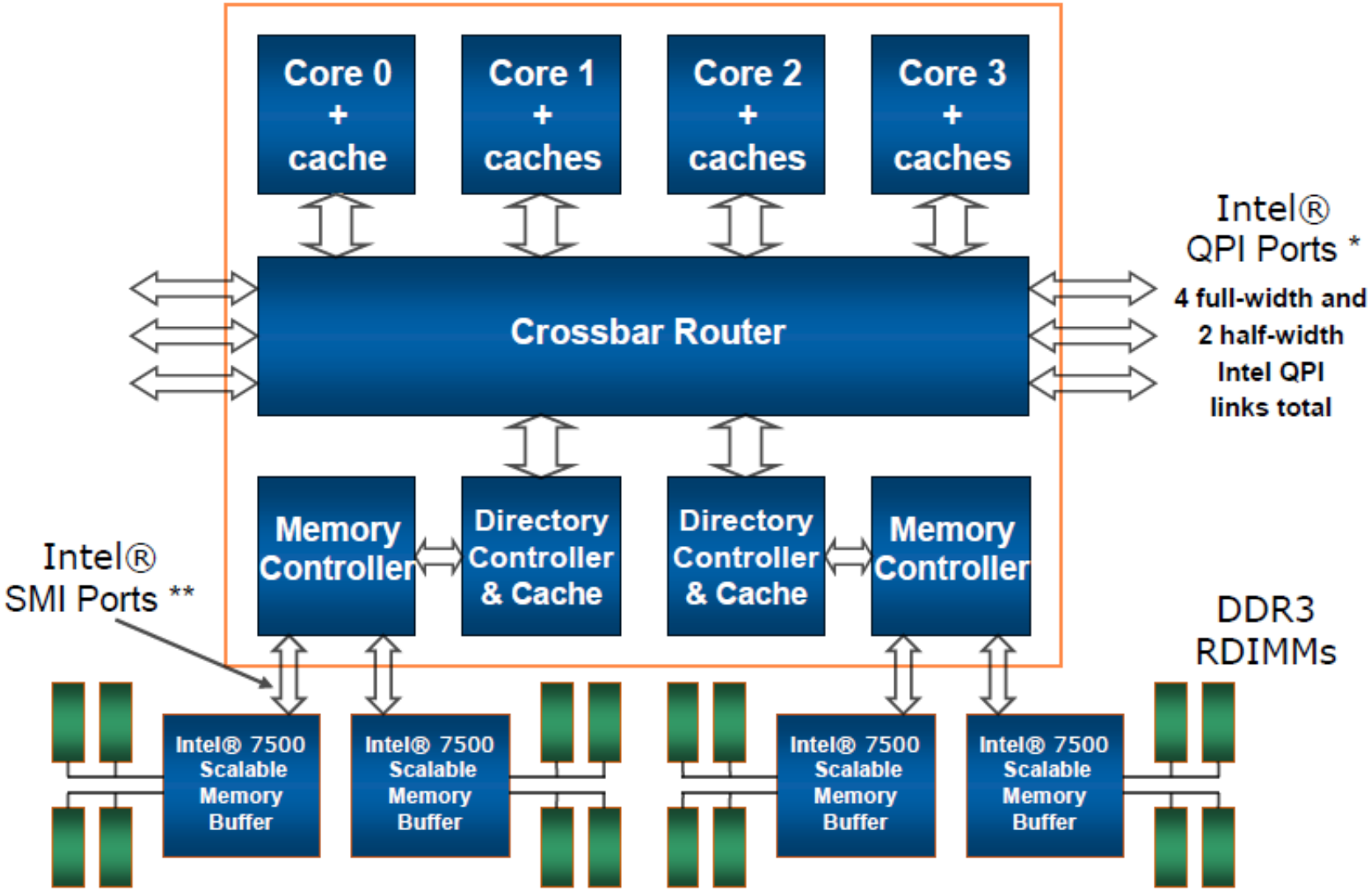
- Same high ILP* core as on the 9000/9100 series processor
  - 6 wide instruction fetch, issue & execution
    - 6 integer units, 2 FP units, 4 memory units, 3 branch units
    - 1 cycle L1 data cache
    - Separate L1I, L1D, L2I, and L2D caches
  - Short 8 stage pipeline – high performance and energy efficiency
  - Predication
  - data prefetching and speculation
- Increased core frequency
- Improved memory hierarchy
  - Increased memory bandwidth
  - 4k/8k/16k page size support in first level data TLB



The cache architecture of Tukwila's cores [47]

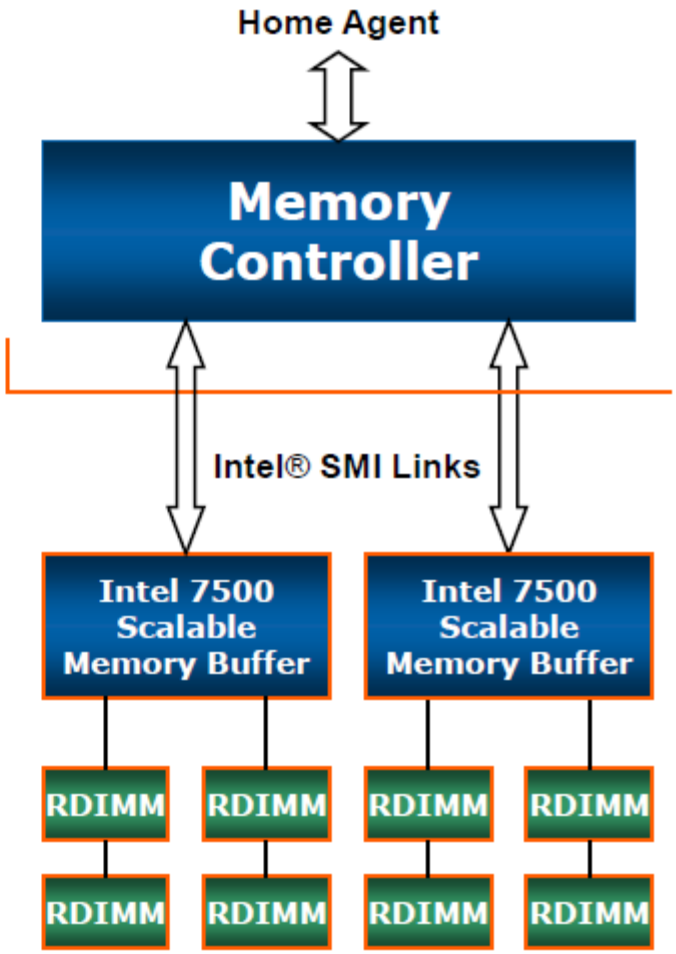


b) Integrated memory controllers-1 [47]

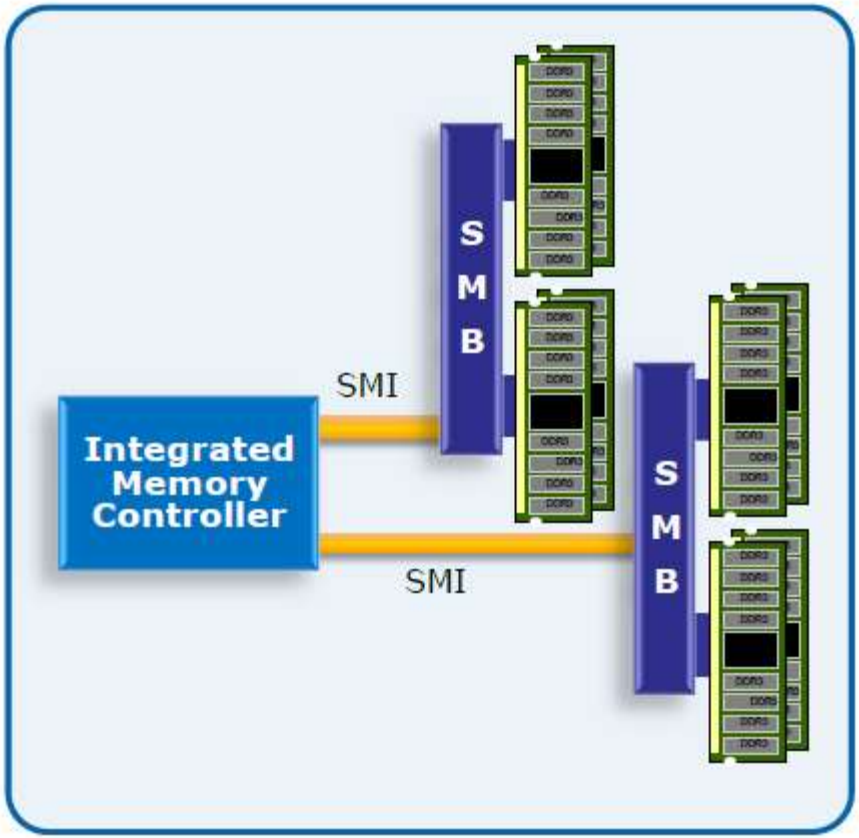


Integrated memory controllers-2 [47]

- Memory Controller
  - Associated with home agent
    - Two memory controllers per processor
  - Each drives two Intel Scalable Memory Interface (Intel SMI) channels in lock-step
    - Higher bandwidth and better error correction capability than single channel
- External Memory Buffer
  - Converts point-to-point SMI channel into two independent DDR-3 busses
  - Improved fan-out for greater memory capacity
    - RDIMMs can be located further away from processor for easier board layout and routing
  - Maintains high bandwidth for large memory capacities
    - Electrical load does not slow down high-speed channels



c) Serial memory connection through SMI links [48]



Integrated Memory Controller (IMC):

- ✓2 IMCs per socket
- ✓High capacity / bandwidth per controller:
  - Up to 8 DIMMs per memory controller
  - 4 DDR Channels for two SMI interface

Scalable Memory Interface (SMI):

- ✓ High speed serial links for maximum bandwidth
- ✓ Run in lockstep mode to minimize latencies & enable RAS

Scalable Memory Buffer (SMB):

- ✓ Memory buffer is on board or memory riser
- ✓ 2 DDR channels & up to 4 DIMMs per buffer

Remark

About at the same time with the Introduction of the Itanium 9300 Tukwila Intel launched also their Nehalem-EX MP servers with SMI links and SMB buffers.

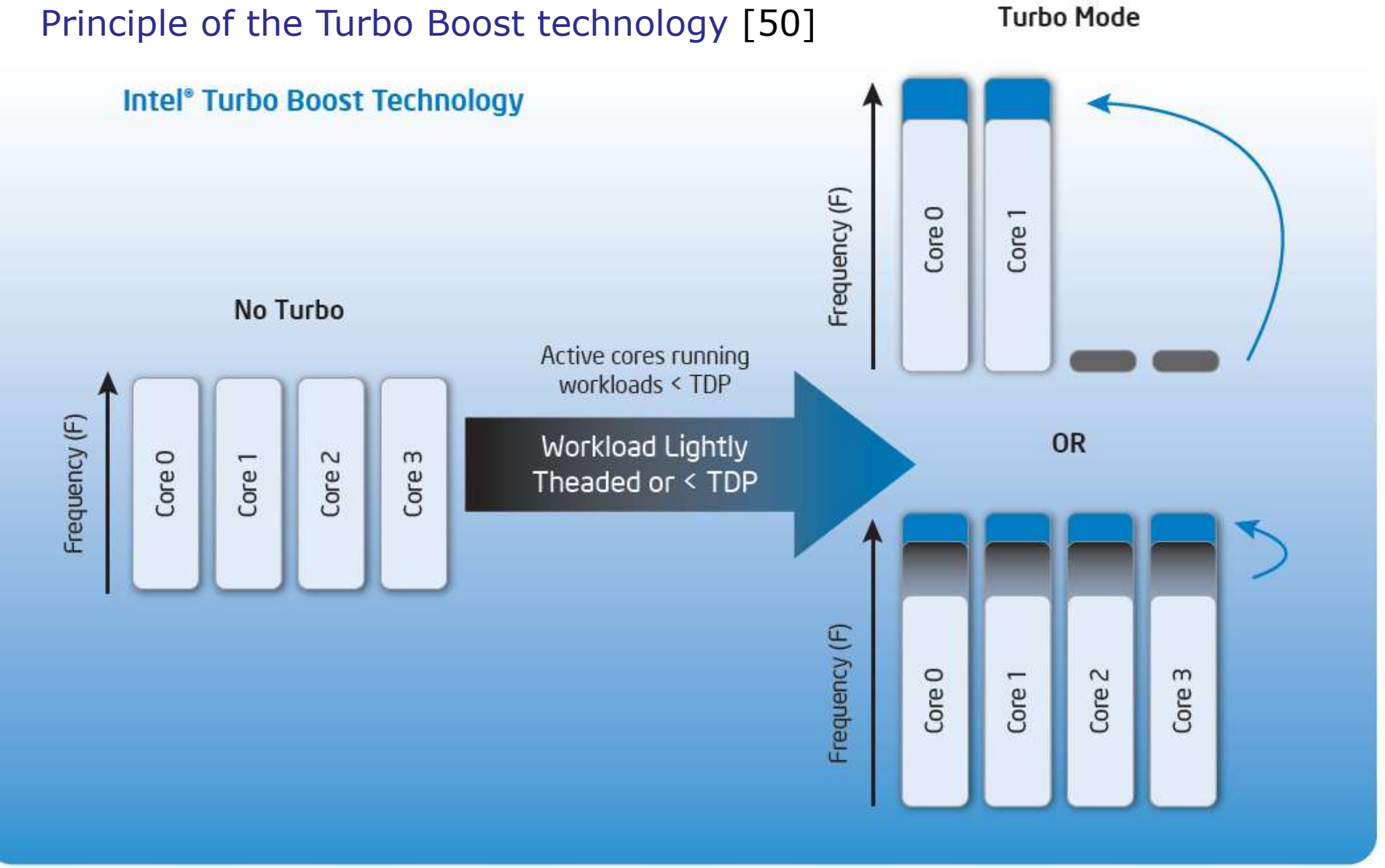
d) Turbo Boost technology

- Intel introduced the Turbo Boost technology in their Tukwila line that debuted previously in their Nehalem desktop line (*Core i7-9xx (Bloomfield) in 11/2008.*
- The Turbo Boost technology was provided only for the 4-core versions with a modest frequency boost of about 10 %, as indicated in the Table below.

Processor	Cores/ threads	Speed	Turbo speed	L3 cache	QPI	TDP	Price
Itanium 9350	4/8	1.73 GHz	1.86 GHz	24 MB	4.8 GT/s	185 W	$3,838
Itanium 9340	4/8	1.60 GHz	1.73 GHz	20 MB	4.8 GT/s	185 W	$2,059
Itanium 9330	4/8	1.46 GHz	1.60 GHz	20 MB	4.8 GT/s	155 W	$2,059
Itanium 9320	4/8	1.33 GHz	1.46 GHz	16 MB	4.8 GT/s	155 W	$1,614
Itanium 9310	2/4	1.60 GHz	N/A	10 MB	4.8 GT/s	130 W	$946

Table: The Intel Itanium 9300 family of server processors [49]

Principle of the Turbo Boost technology [50]



Turbo Boost uses the available power headroom in the processor package [50]

e) QPI-based SMP interconnect

Tukwila provides 4x full width + 2x half-width 4.8 Gb/s QPI links for building SMPs, as shown below.

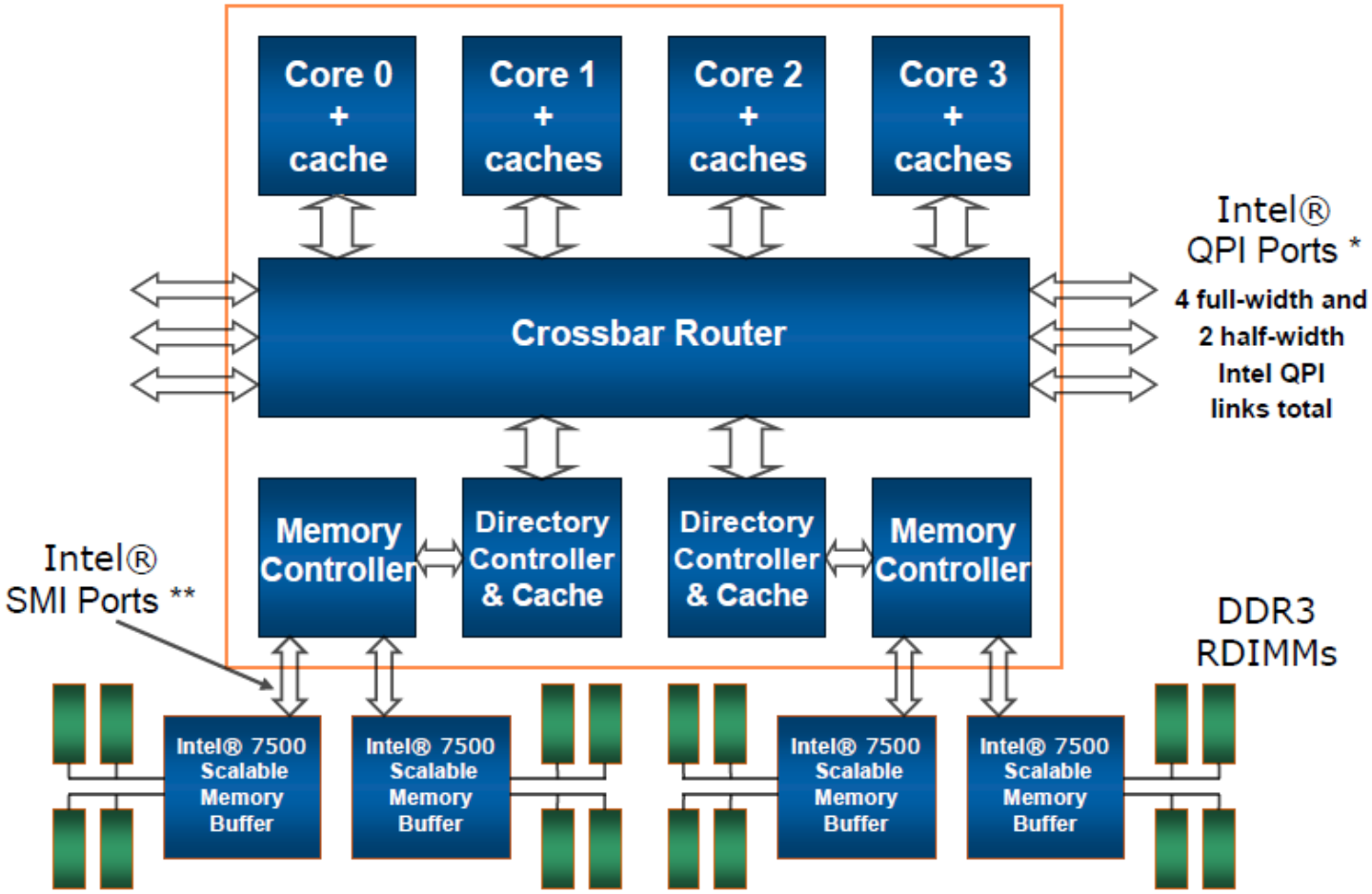
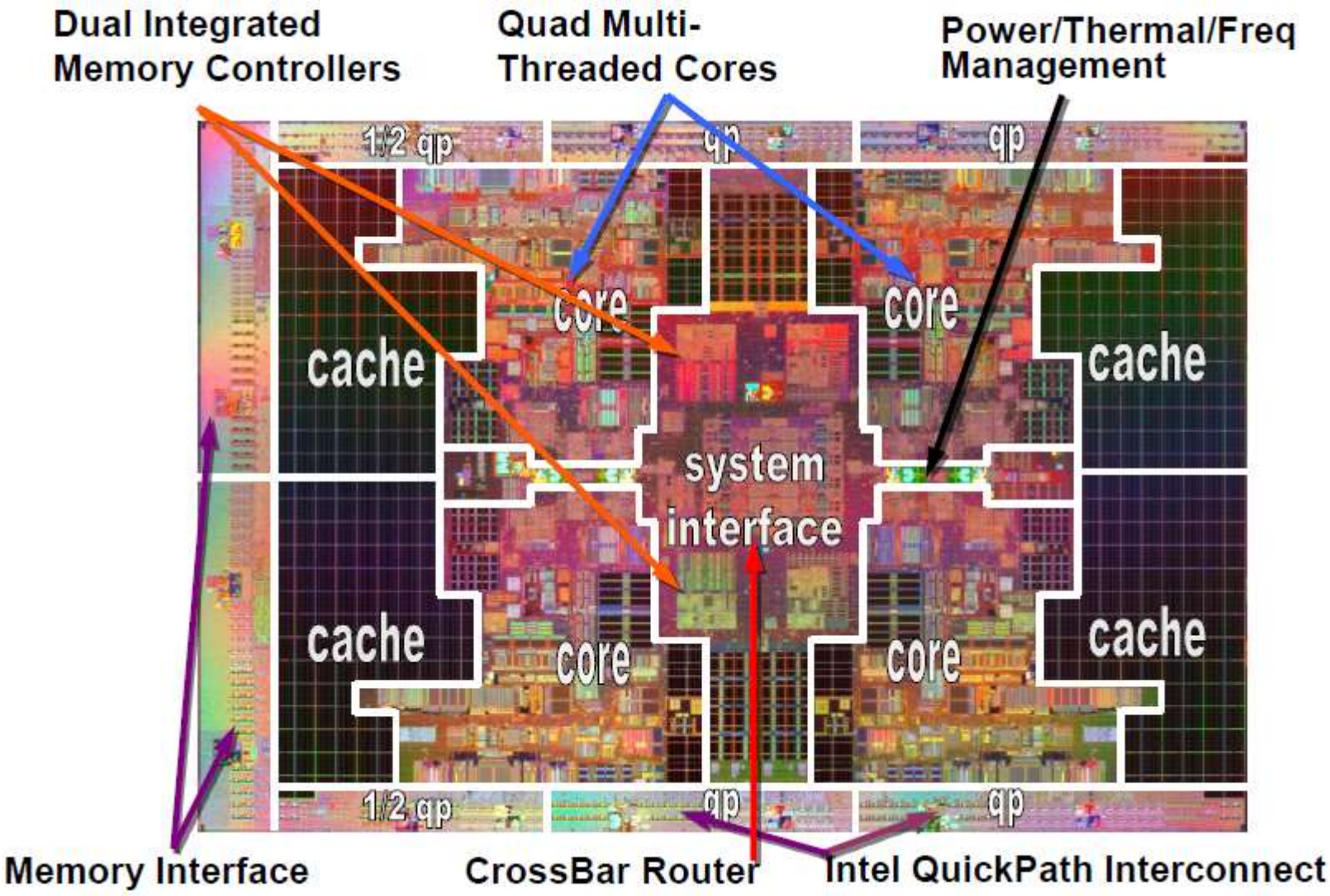
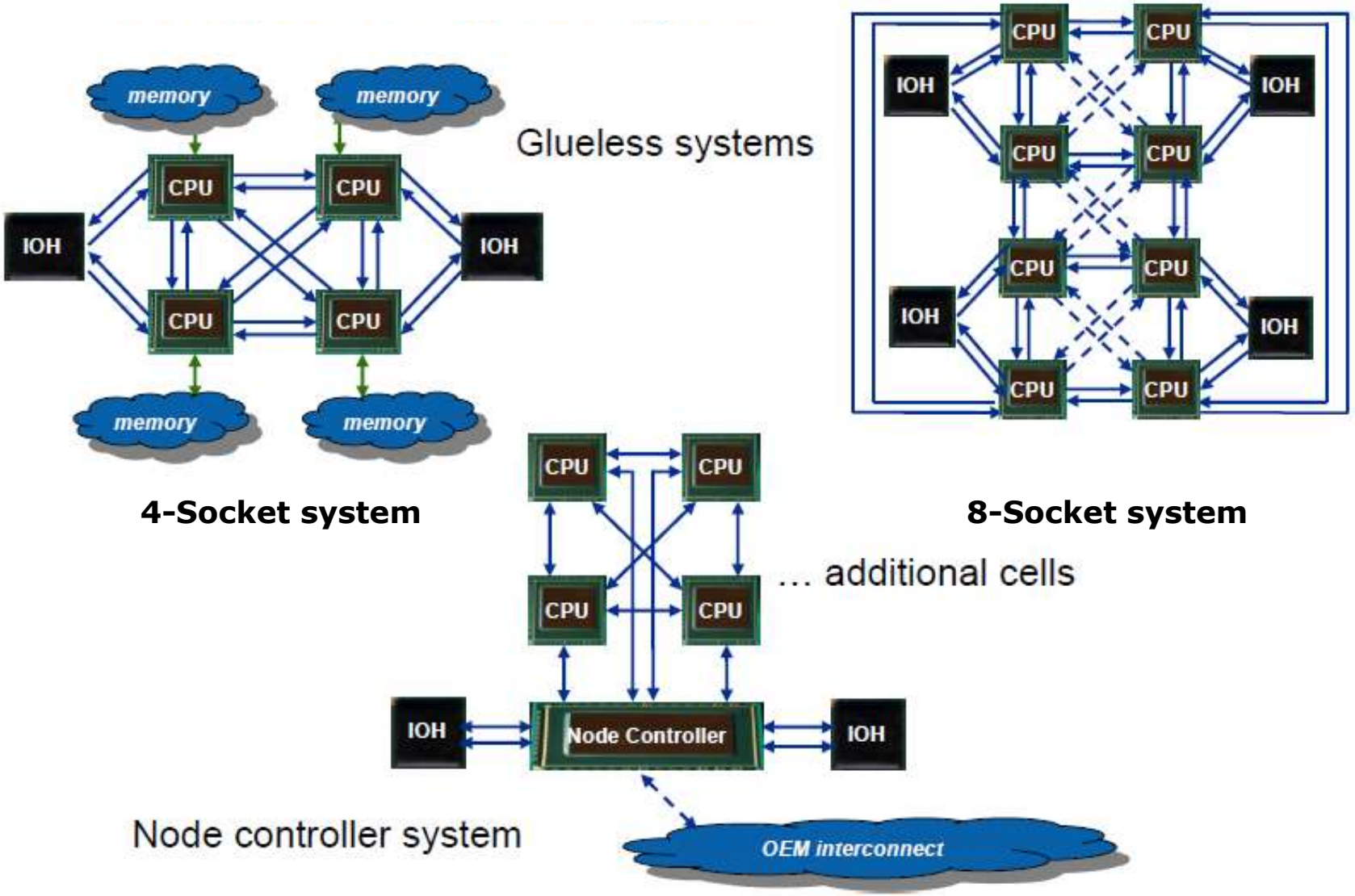


Figure: High level block diagram of Tukwila [47]

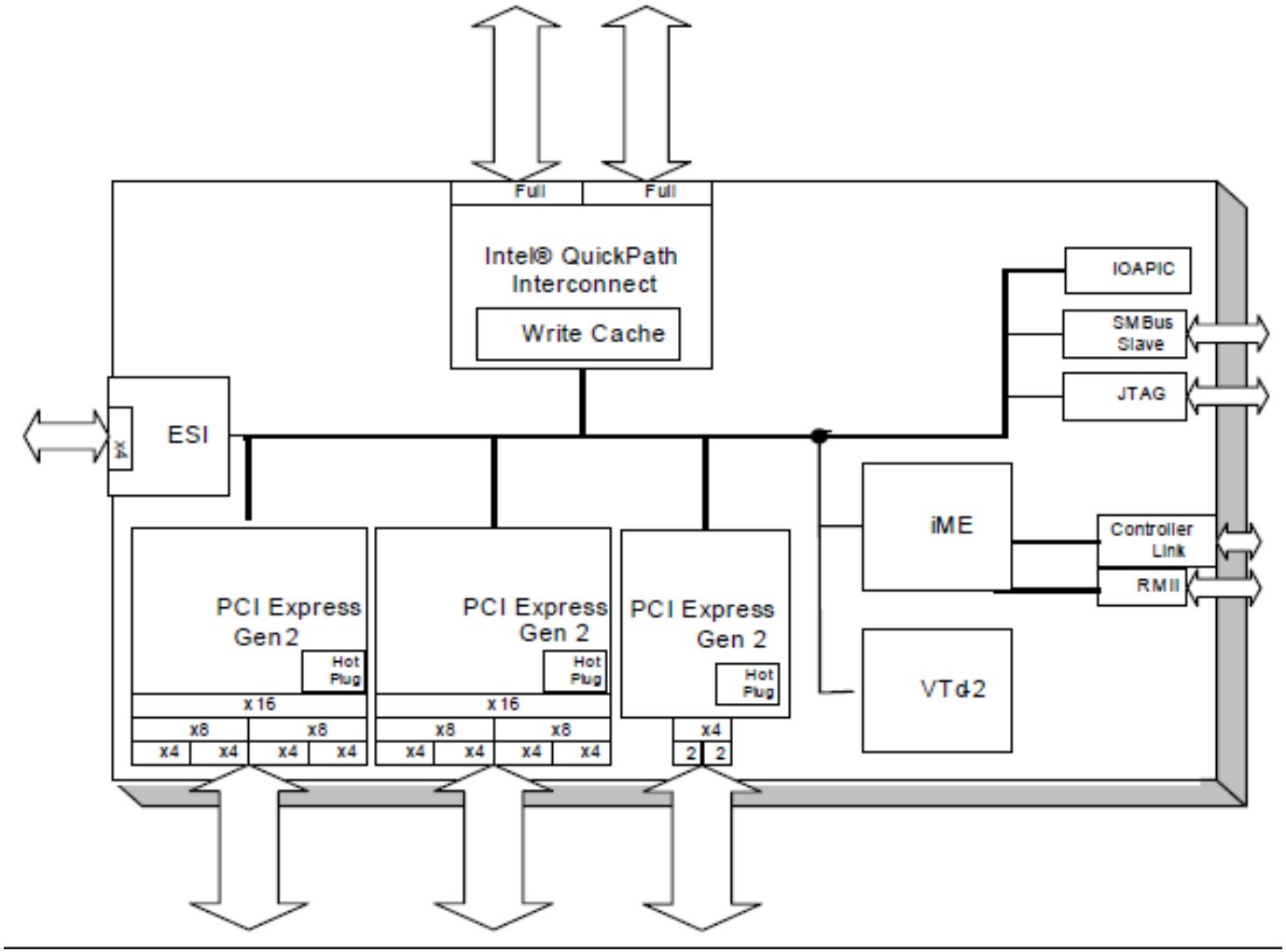
Die photo of Tukwila [47]



Example topologies based on the QPI coherent bus [47]



High-level block diagram of the 7500 chipset [51]



f) On-die directory based cache coherency

Tukwila supports on-die directory based cache coherency, as shown below.

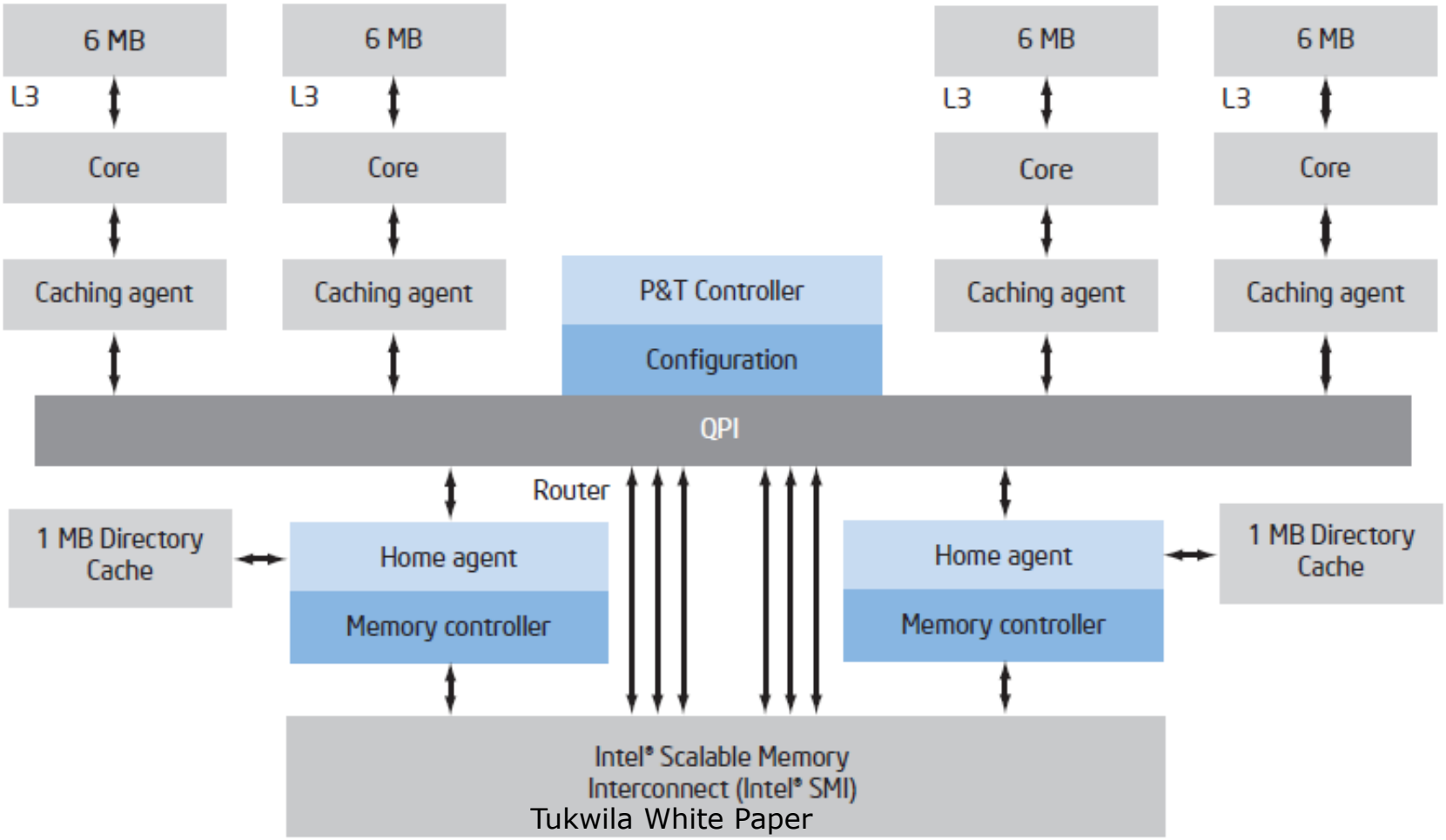


Figure: Block diagram of Tukwila [52]

g) Vastly enhanced RAS features-1 [52]

RAS Capability	Intel® Itanium® Processor 9100 Series	Intel® Itanium® Processor 9300 Series
Processor Core		
Extensive Error Protection/Correction On-core structures and system interface: • ECC, parity and/or SER hardened latches and registers are used to avoid, detect and correct errors.	•	Enhanced
On-cache structures: • ECC is used to detect and correct errors. • Intel® Cache Safe Technology is used to disable failed cache lines for improved availability.	L3 only	L2, L3 and Directory
Processor Socket		
Advanced Error Protection/Correction on the Processor Links • **Dynamic Link Rerouting:** Sustains uninterrupted operation if an Intel® QuickPath Interconnect or Intel® Scalable Memory Interconnect (Intel® SMI) link physically fails.	•	Enhanced
Processor Onlining/Offlining: A processor can be functionally enabled or disabled without downtime to adjust available resources or to map out a failed component.	OEM-based[c]	Native support
Processor Hot Plug: A processor can be physically added, removed or replaced without downtime for system upgrades or to replace a problematic component.	OEM-based[c]	Native support
Intel® Virtualization Technology – processor cores: Hardware-based virtualization support improves ability to implement transparent workload migration to optimize resource utilization and simplify failover.	Intel VT-i	Intel VT-i2

Vastly enhanced RAS features-2 [52]

Memory Error Correction mechanisms include: ▪ **Memory ECC Support:** Automatically detects and corrects all single-bit errors and most double bit stored errors (uncorrectable errors are detected and reported). Errors in up to eight consecutive bits can be corrected. ▪ **Single Device Data Correction:** Automatically corrects multi-bit errors on a single DRAM device; can map out a failed device and continue correcting single-bit errors. ▪ **Dual Device Data Correction:** Automatically corrects multi-bit errors on two DRAM devices; can map out two failed DRAM devices and continue correcting single-bit errors.	OEM-based[c] or Intel chipset required	Native support
	OEM-based[c]	Native support
Memory Channel Protection: Includes three levels of protection, Cyclic Redundancy Check (CRC) to detect and repair transient errors; physical layer reset for persistent errors; and lane failover if the reset fails.	OEM-based[c]	Native support
Memory Scrubbing: Memory is monitored to correct errors, which protects correctable errors from accumulating and becoming uncorrectable. Performed automatically and periodically (Patrol) and also at the request of the OS (Demand).	OEM-based[c]	Native support
Memory DIMM sparing: Firmware copies data from a failing DIMM to a spare DIMM on the same memory channel, and maps out the failed component to enable uninterrupted operation.		▪
Memory Migration: Firmware copies data from a failing DIMM and migrates it to a DIMM on another memory controller of the same or another processor.	OEM-based[c]	Native support
Memory Mirroring: A backup copy of main memory can be maintained for very high-reliability error correction (if used, requires twice the memory).	OEM-based[c]	Native support
Memory Onlining/Offlining: One or more DIMMS or memory riser cards can be functionally enabled or disabled without downtime to adjust available resources or to map out a failed component.	OEM-based[c]	Native support
Memory Hot Plug Support: Memory components (DIMMs) can be physically added, removed or replaced without downtime. Includes OS-visible and OS-transparent capabilities.	OEM-based[c]	Native support[d]
Memory Thermal Protection includes: ▪ **Closed Loop thermal throttling:** Memory channel activity can be reduced (or fan speed increased) when the temperature of a DIMM exceeds a preset level. ▪ **Open Loop Thermal Throttling:** Memory channel activity can be reduced (or fan speed increased) when the number of memory commands per DIMM exceeds a configurable limit over a configurable time interval.		▪

RAS enhancements of the cache subsysten - Tukwila vs. Montvale [46]

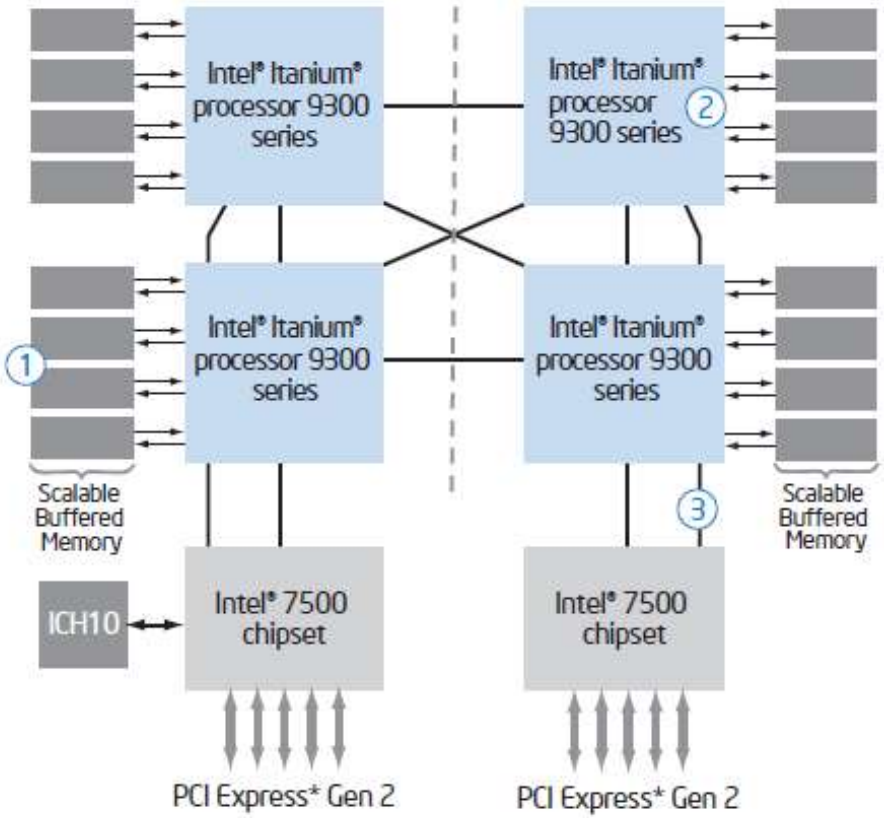
Cache Array	Size	ECC/ Parity	Phase/ Cycle Access	Voltage Domain	Ramcell Sleep	Redun-dancy	CacheSafe Technology
L3	**6MB/ Core**	ECC	Cycle	Vcache	**Yes**	**Yes (Block)**	Yes
L3 Tag	**250KB/ Core**	ECC	Phase	**Vcache**	No	**Yes (Entry)**	No
L2 Instr	**512KB/ Core**	Parity	Cycle	**Vcache**	No	**Yes (Block)**	**Yes**
L2 Instr Tag	**20KB/Core**	Parity	Phase	**Vcache**	No	No	No
L2 Data	256KB/Core	ECC	Phase	**Vcache**	No	**Yes (Row)**	**Yes**
L2 Data Tag & L1	Various	Various	Various	Vcore	No	No	No
Directory Cache	**1.9MB/Die**	**Parity**	**Cycle**	**Vcache**	**No**	**Yes (Column)**	**Yes**

Changes from Montvale in gray

RAS features of Tukwila – Overview [52]

1. Memory RAS

- DRAM Protection (ECC, SDDC, DDDC),
- Memory Scrub (Patrol and Demand)
- Memory Thermal Protection (CLTT and OLTT)
- Memory Channel Protection (Retry, Reset, Lane Failover)
- Memory Migration
- DIMM Sparing
- Memory mirroring
- Memory Channel Hot Plug



2. Processor RAS

- ECC and Parity protection
- SE Hardened Latches and Registers
- Intel® Cache Safe technology (L2I, L2D, L3, directory cache)
- Processor Hot Plug
- Advanced Machine Check Architecture with new CMCI support

3. Intel® QuickPath Interconnect RAS

- Detection/Correction (CRC, Retry, Reset, Lane Failover)
- Hot Plug Links (supports Hot Plug for I/O Hub and PCIe cards)
- Domain Partitioning
- Intelligent Error Management

h) Multi-threading enhancements [53]

Tukwila provides multi-threading enhancements to hide more stall cycles

– Improved thread switch events

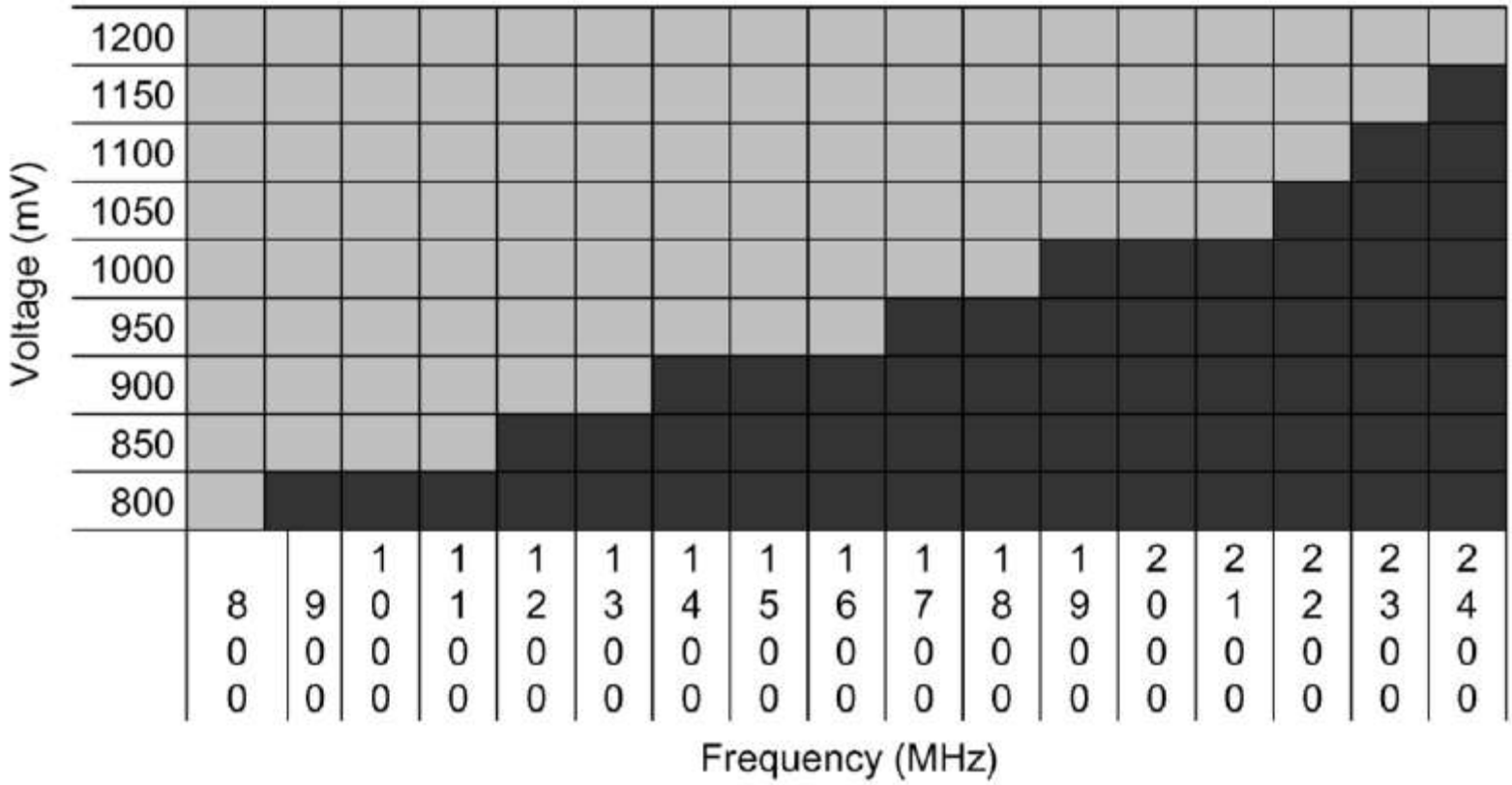
- Allow thread switches on pipeline stalls that are not necessarily L3 cache demand misses (e.g. secondary misses after a prefetch).
- Switch on semaphore release

– Improved thread switch algorithms

- Thread switch decision is based on “urgency counters” which are based on hardware events.

Changes to urgency update logic have improved multi-threading performance.

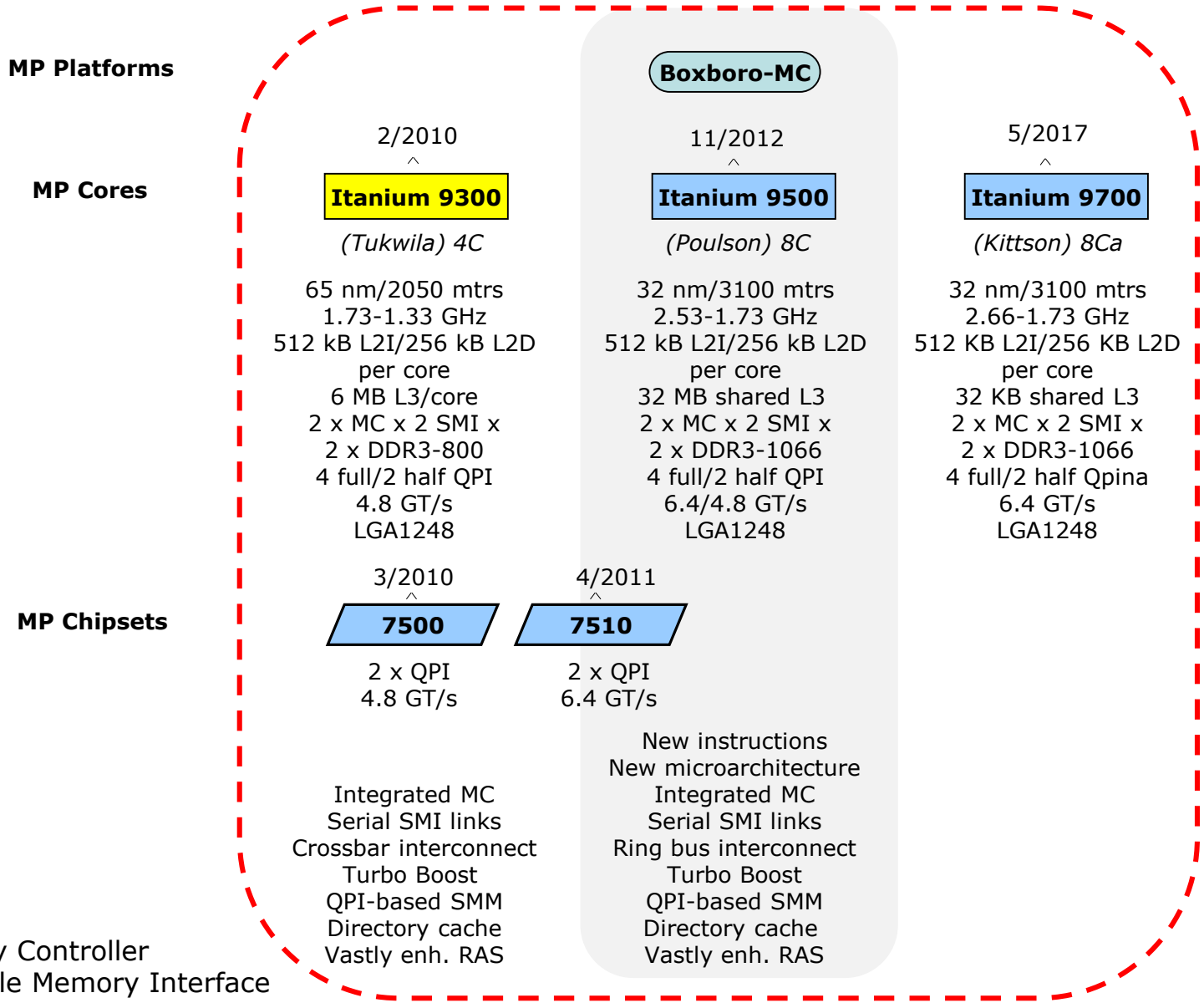
The Shmoo diagram of Tukwila [46]



Core shmoo at $T_j = 80\,°C$.

7.3 The 8-core Poulson (Itanium 9500) line

7.3 The 8-core Poulson (Itanium 9500) line-1



MC: Memory Controller
SMI: Scalable Memory Interface

The 8-core Poulson (Itanium 9500) line-2

- Introduced: 11/2012
- Manufactured with 32 nm feature size, 3100 mtrs
- Drop-in replacements for previous Itanium 9300 (Tukwila) processors.

Key enhancements of the Poulson (Itanium 9500) line

a) 8 cores
b) new instructions
c) significantly raised clock frequencies
d) new microarchitecture design
e) on-die ring interconnect.

a) 8 cores [54]

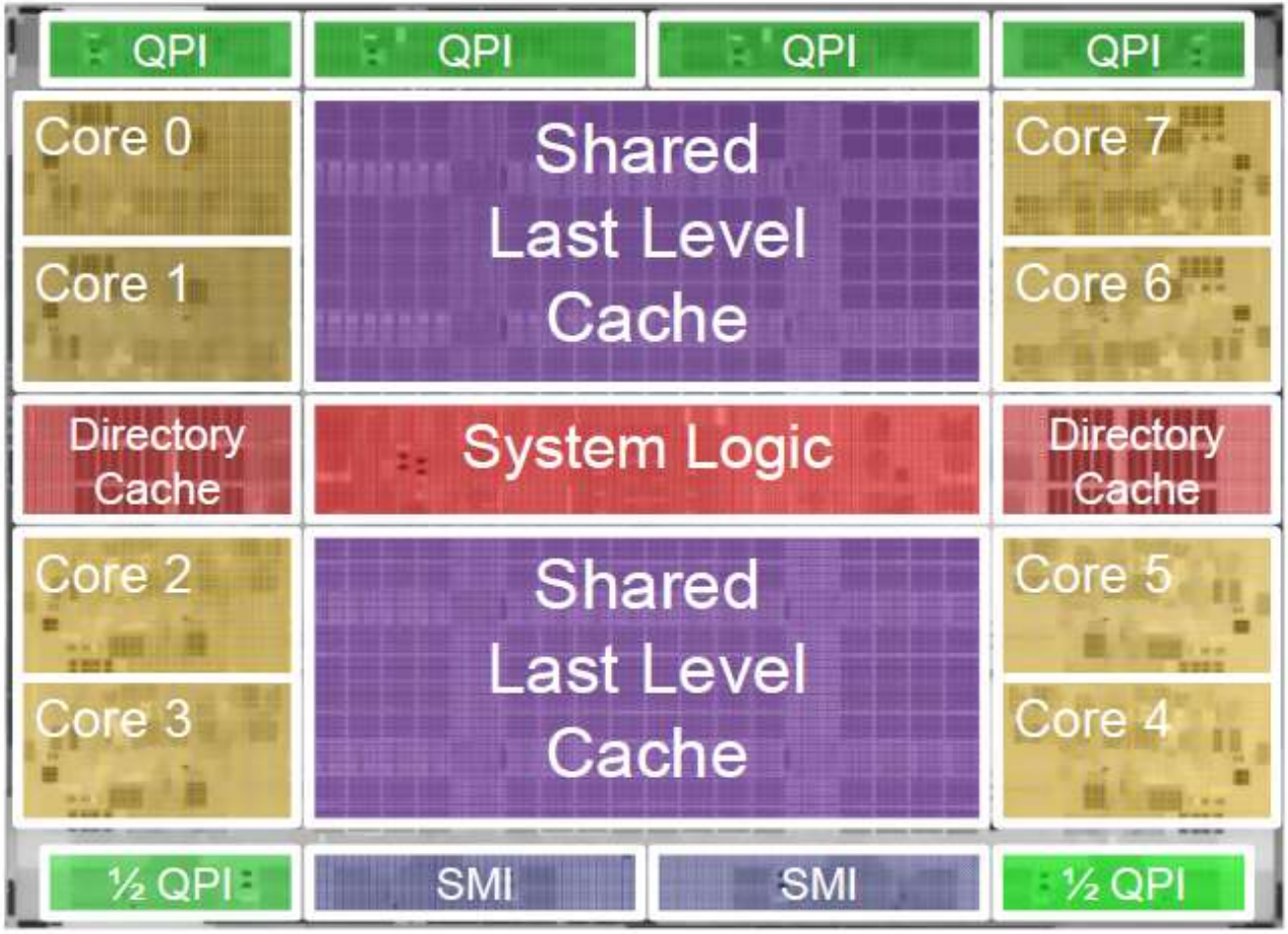
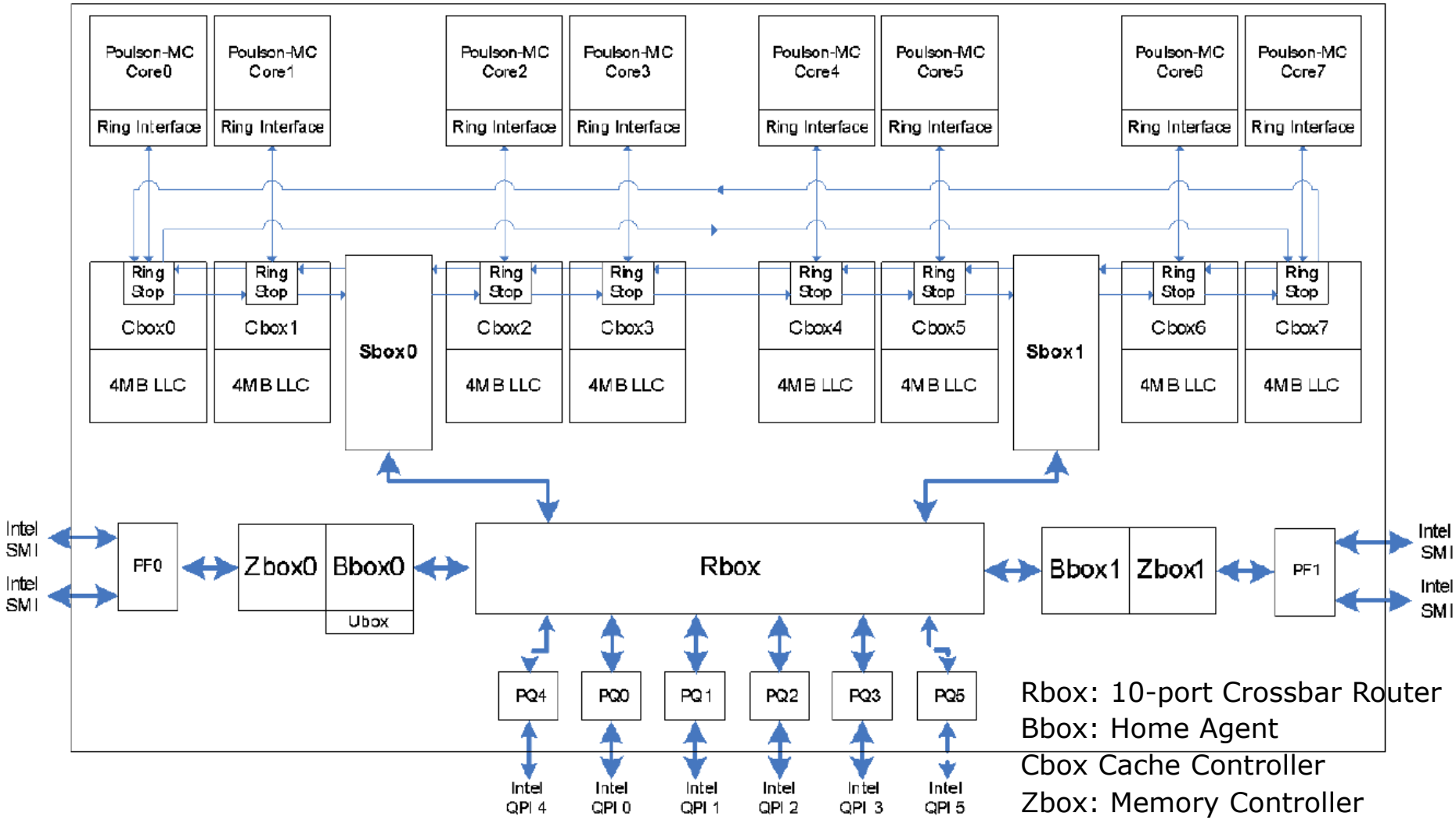


Figure: Block diagram of the Poulson processors [54]

Intel Itanium 9500 and 9700 series block diagram [71]



Rbox: 10-port Crossbar Router
Bbox: Home Agent
Cbox Cache Controller
Zbox: Memory Controller

7.3 The 8-core Poulson (Itanium 9500) line (4)

b) Significantly raised clock frequencies

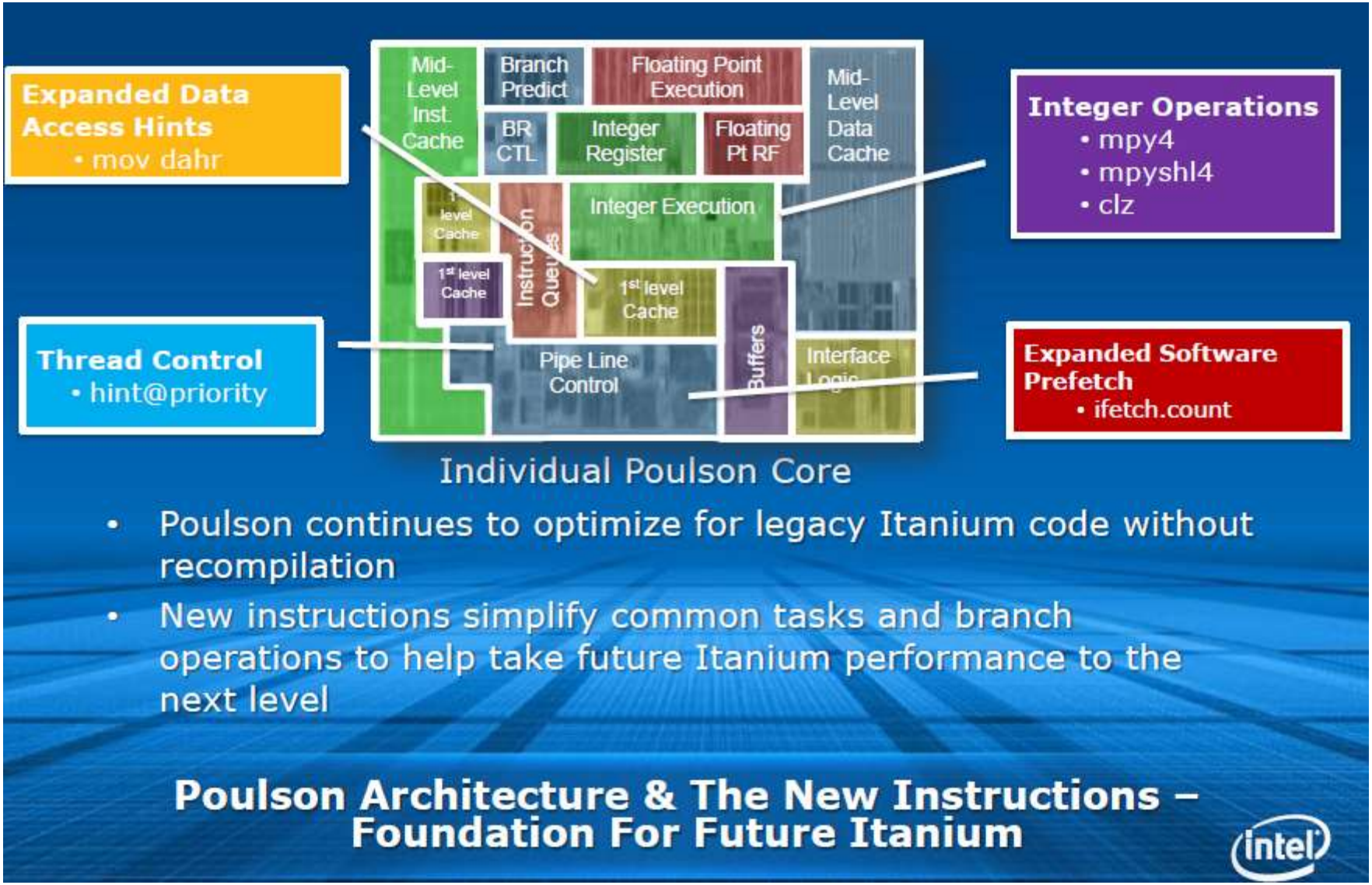
- Tukwila had a max. core frequency of 1.73 GHz.
- By contrast, Poulson models have significantly higher clock speeds, as the Table below demonstrates.

Intel® Itanium® Processor 9500 Series Product SKUs

PROCESSOR NUMBER[Δ]	OPTIMIZED FOR	CORES / THREADS	LAST LEVEL CACHE (L3)	POWER	CPU FREQUENCY
Intel® Itanium® Processor 9560	Performance	8 / 16	32 MB	170 W	2.53 GHz
Intel® Itanium® Processor 9550	Performance per core	4 / 8	32 MB	170 W	2.4 GHz
Intel® Itanium® Processor 9540	Price Performance	8 / 16	24 MB	170 W	2.13 GHz
Intel® Itanium® Processor 9520	Value	4 / 8	20 MB	130 W	1.73 GHz

Table: Main features of Poulson models at announcement [57]

c) New instructions [55]



7.3 The 8-core Poulson (Itanium 9500) line (6)

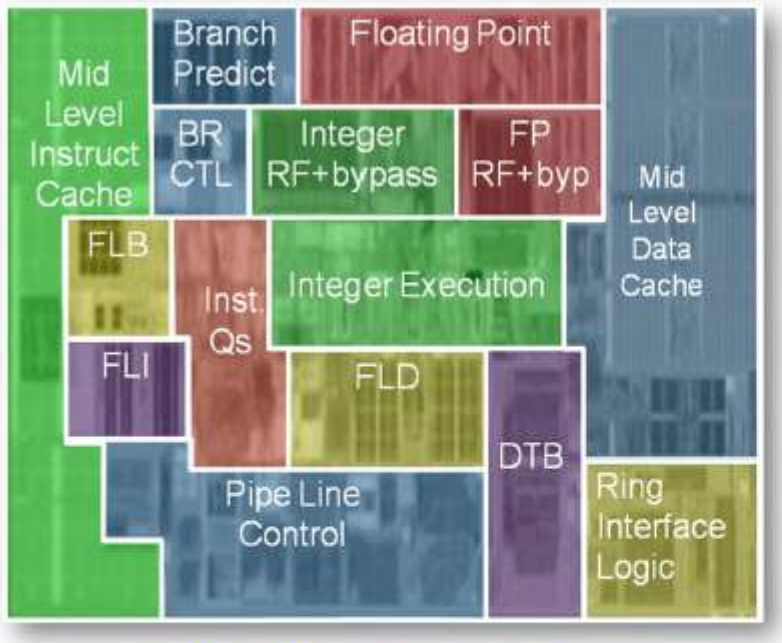
d) New microarchitecture design

- The Poulson is the first comprehensive redesign of the microarchitecture since the Itanium 2 McKinley [54].
- Here we do not go into the exhausting details of the redesign but below we expose only a few aspects of it.
- A detailed description of the new microarchitecture can be found in [31].
- Subsequently discussed features of Poulson's redesigned microarchitecture:

d1) New pipeline design
d2) Multi-threading improvements
d3) Core pairs with separate power planes and voltage regulations
d4) Enhanced cache hierarchy
d5) Increased QPI speed
(instead of 4.8 GT/s 6.4 GT/s)
d6) RAS improvements

- 12 execution pipelines instead of 11
- 12-wide back end instead of 6-wide
- replay based pipeline design
- core pairing
- 6.4 GT/s QPI

The core architecture of Poulson [54]



Individual Poulson Core

Key Architectural Advances

- *New Data and Instruction Pipelines*
- *New Floating Point Pipeline*
- *New Instruction Buffer*
- *Double max execution width*
  - *From 6 to 12 wide*

Derived Benefits

- *Increased instruction throughput*
- *Improved performance/watt*
- *Improved RAS coverage*
- *Core optimized for future technologies*

Increased performance, power reduction and reliability

Core architecture enables futures IPF processor designs

7.3 The 8-core Poulson (Itanium 9500) line (8)

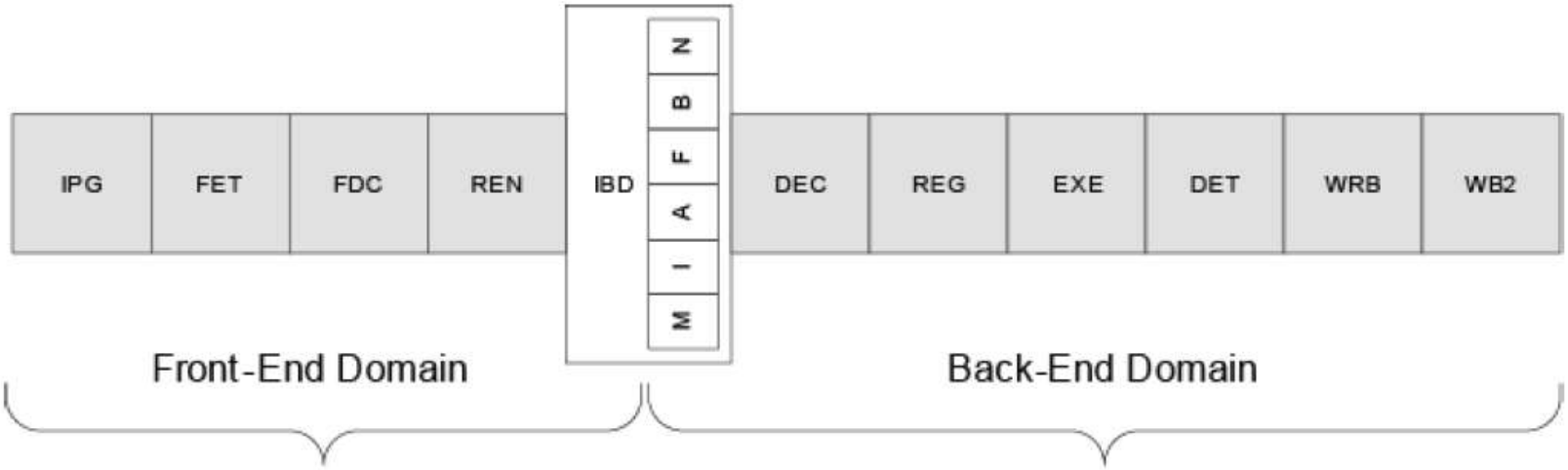
d1) New pipeline design-1

From the enhanced pipeline design we point out only two features, as follows:

- Lengthening the pipeline from eight to eleven stages,
- Doubling the issue rate from 6 instructions to 12 istructions and
- Introducing replay based pipeline design.

Lengthening the pipeline from eight to eleven stages

The new pipeline design has the following 11 stages instead of 8 stages as used in the previous McKinley design, as indicated below.



IPG – Instruction *Pointer Generate*
FET – Instruction *Fetch*
FDC – *Front-end Decode*
REN – Register Rename

IBD – Instruction Buffer and Dispersal
DEC – Instruction *Decode*
REG – *Register* Access
EXE – Instruction Execute
DET – *Detect Exceptions*
WRB – *Writeback* Commit
WB2 – *Writeback-2 Retire*

Figure: The 11-stage pipeline of Poulson [56]

Presumably, the longer pipeline contributed significantly to raising the clock speed.

Doubling the issue rate from 6 instructions to 12 instructions-1 [56]

In Poulson there are 12 issue ports instead of 11 in the McKinley design.

In concert, the issue rate has been doubled from 6 instructions (two bundles) to 12 instructions (four bundles).

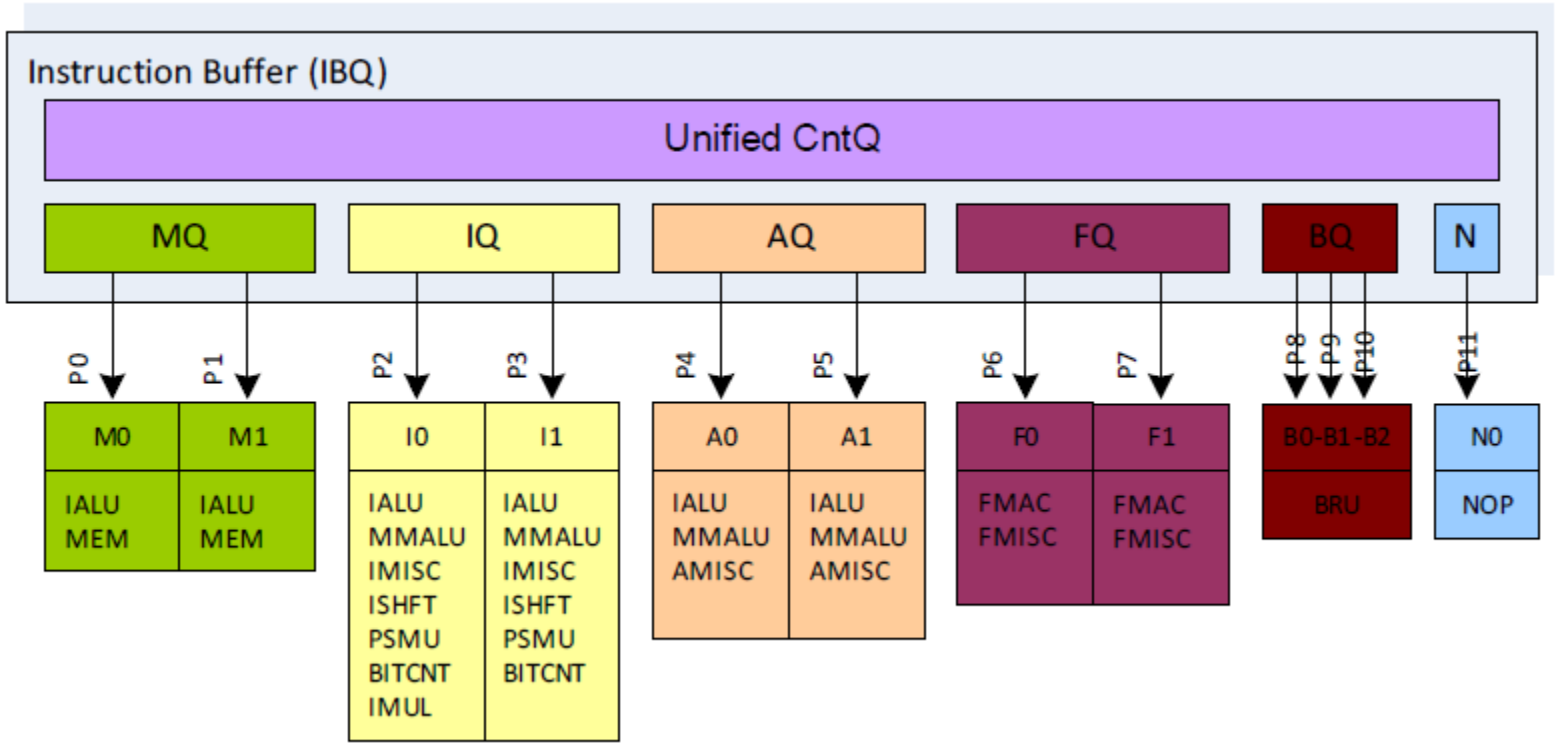


Figure: Redesigned issue ports of Poulson [56]

7.3 The 8-core Poulson (Itanium 9500) line (11)

Some features of the execution pipelines [56]

Port Type	Pipelines	Functional Capabilities
M-port	M0, M1	1-cycle ALU, Memory Address support
I-port	I0	1-cycle ALU, 2-cycle MM ALU, Integer Shifter, MM Shifter, Misc I-unit, Integer Multiplier
	I1	1-cycle ALU, 2-cycle MM ALU, Integer Shifter, MM Shifter, Misc I-unit
A-port	A0, A1	1-cycle ALU, 2-cycle MM ALU, Misc A-unit
F-port	F0, F1	6-cycle Fmac unit, 6-cycle Fmisc unit
B-port	B0, B1, B2	Branch resolution unit
N-port	N0	Handles up to 12 nop, brp instructions

Doubling the issue rate from 6 instructions to 12 instructions-2 [31]

Here we point out that while instruction issue and execution is done at the instruction level, instruction retirement is still performed at bundle granularity, i.e. Instructions that have completed will wait in the instruction queues until the whole bundle can retire.

Introducing replay based pipeline design

- Previous Itanium processors had a stall based pipeline design, by contrast Poulson has a replay based pipeline microarchitecture.
- Both the front end and the back end pipelines have their own pipeline control providing replays, flushes and stalls to address various pipeline hazards.

The Figure below shows the main pipeline's control mechanisms.

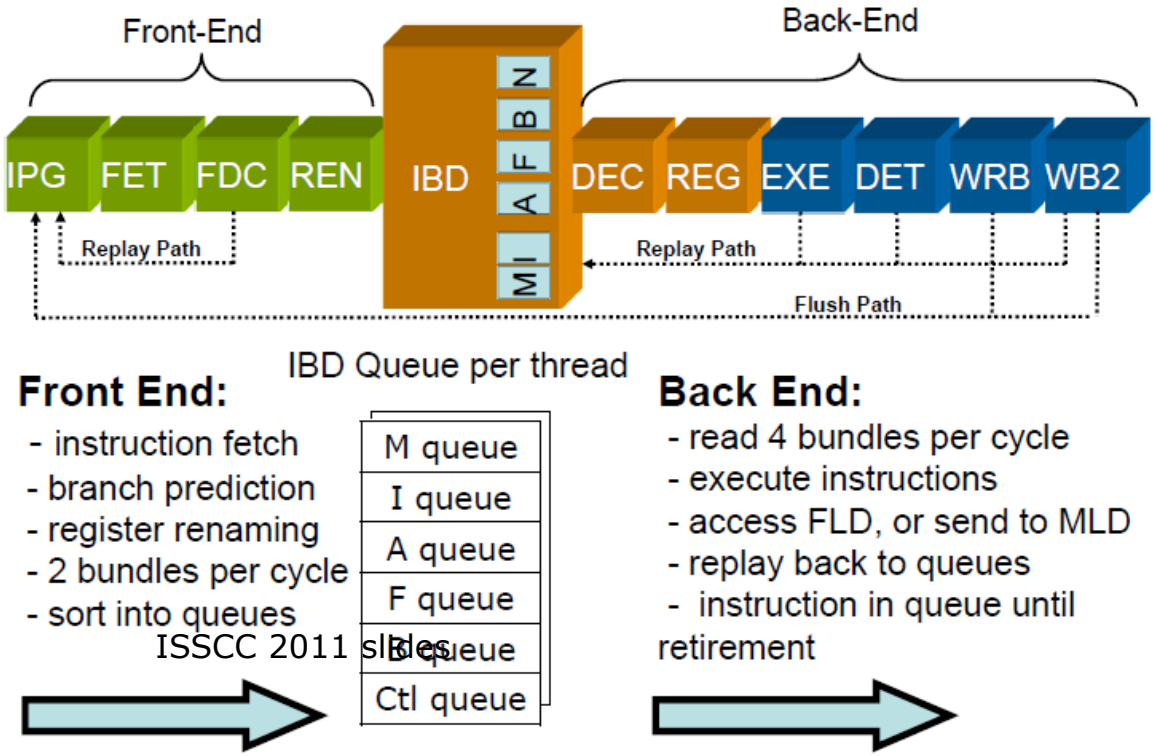
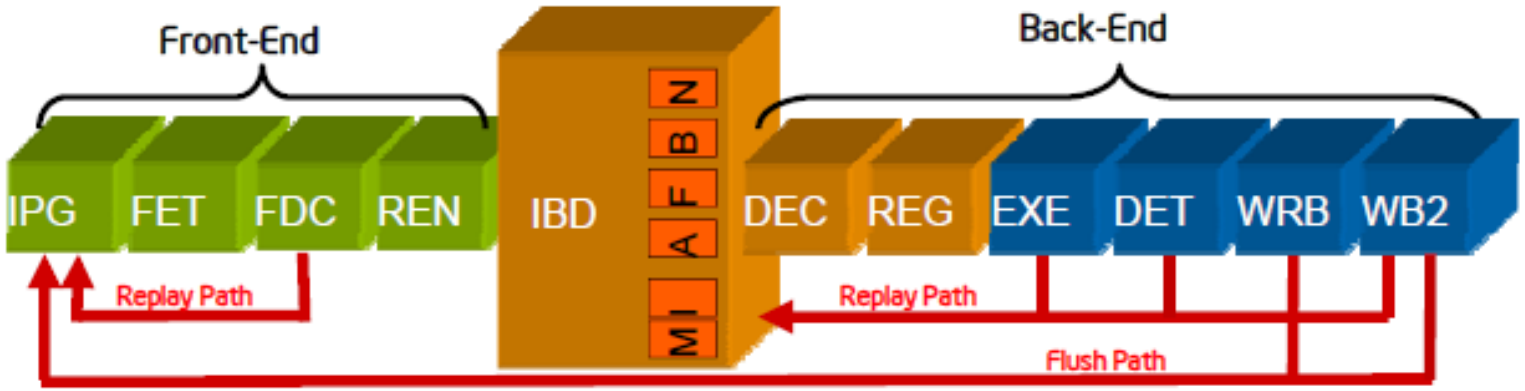


Figure: Poulson's main pipeline's control mechanisms [54]

RAS features covered by instruction replay [58]

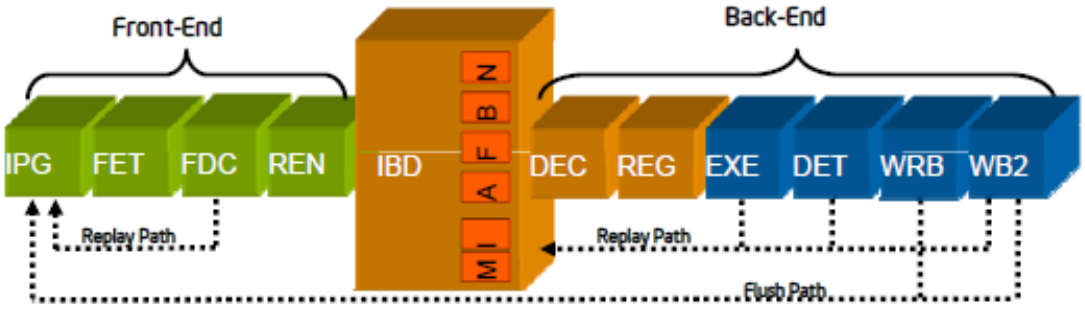


IBD: Instruction Buffer

- Front-end data integrity errors (e.g., rename parity error) resolved with front-end replay path and instruction re-fetch
- Back-end data integrity errors (e.g., FP residual error) resolved by reissuing instruction from IBD
- IBD parity error resolved by flush of full pipeline and instruction re-fetch

d2) Multi-threading improvements [58]

- Dual-domain multi-threading combines elements of SOEMT (Switch-on-event) and SMT (Simultaneous) multi-threading



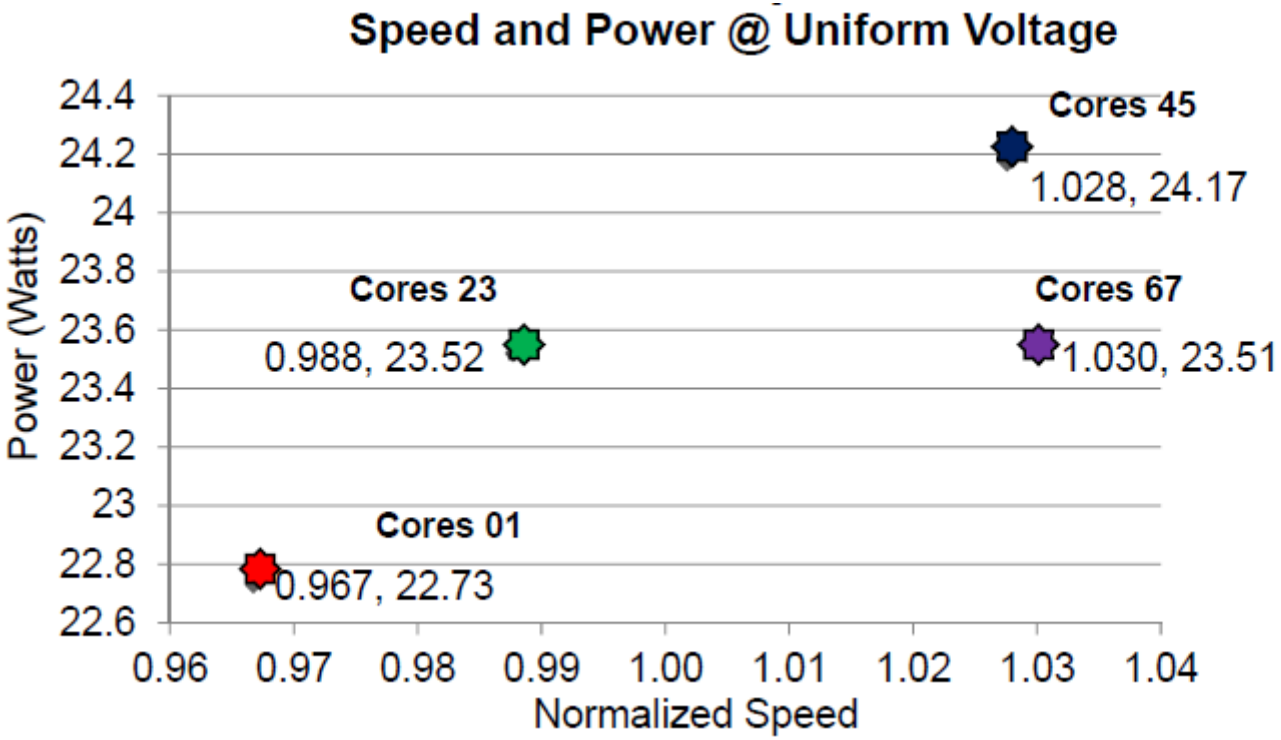
- Front-end and back-end can operate on different threads
- IBD stores 96 instructions per thread (96x2 queue entries)
  - Maximizes efficiency of concurrent threads to drive 12-wide issue
- Thread change only drains back-end pipeline and starts with full IBD
  - Minimizes thread switch costs of previous Itanium MT implementations
- Front-end and back-end can switch threads on different conditions

d3) Core pairs with separate power planes and voltage regulations [31]

- Transistors fabricated with a feature size of 32 nm vary significantly on a large die in terms of speed and power consumption.

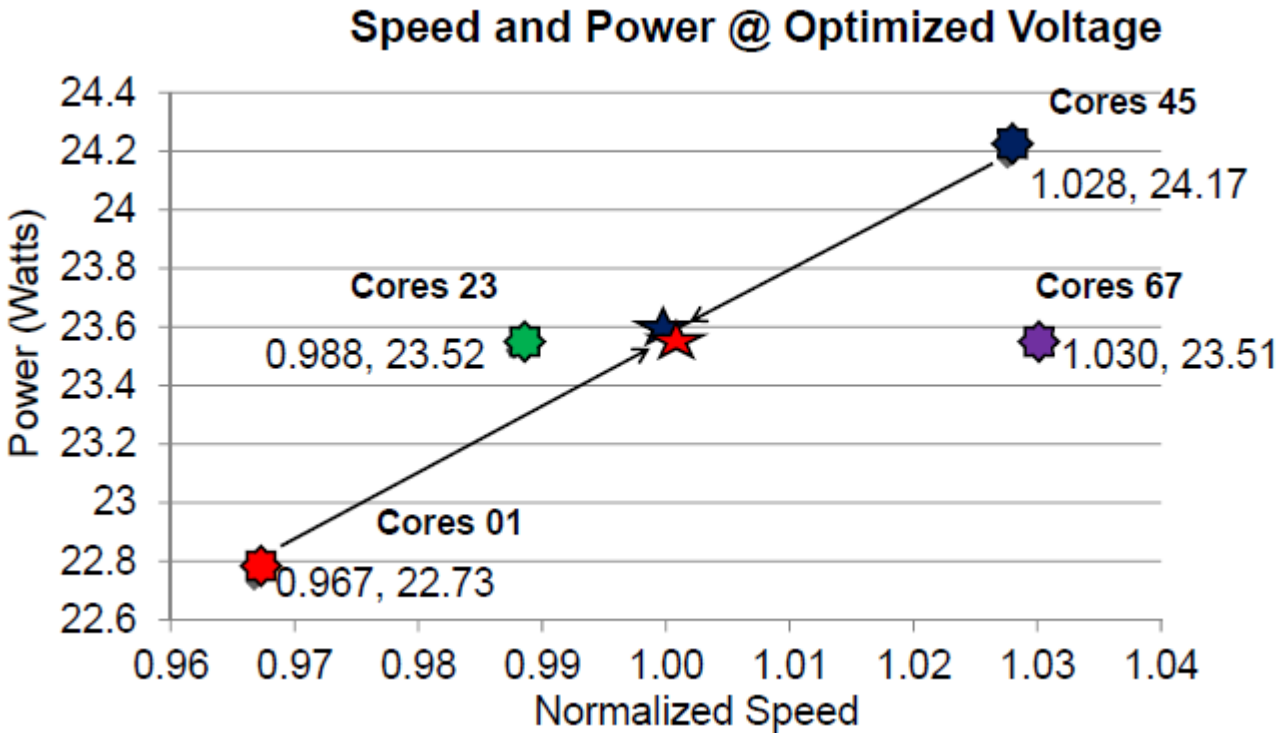
  So some cores on a die may be 'slow and cool', while others might be 'fast and hot'.
- In Poulson there are four core pairs, and each has a separate power plane and voltage regulation (with a target range of 0.85-1.2V).
- After manufacturing, Intel adjusts the voltage down on fast cores, and raises the voltage for slow cores to reach the same target frequency on all eight cores, as demonstrated in the next Figures.
- Using core pairs with four separately regulated core voltage supplies increases frequency by about 5%, with no additional power consumption.

Core pair optimization-1 [54]



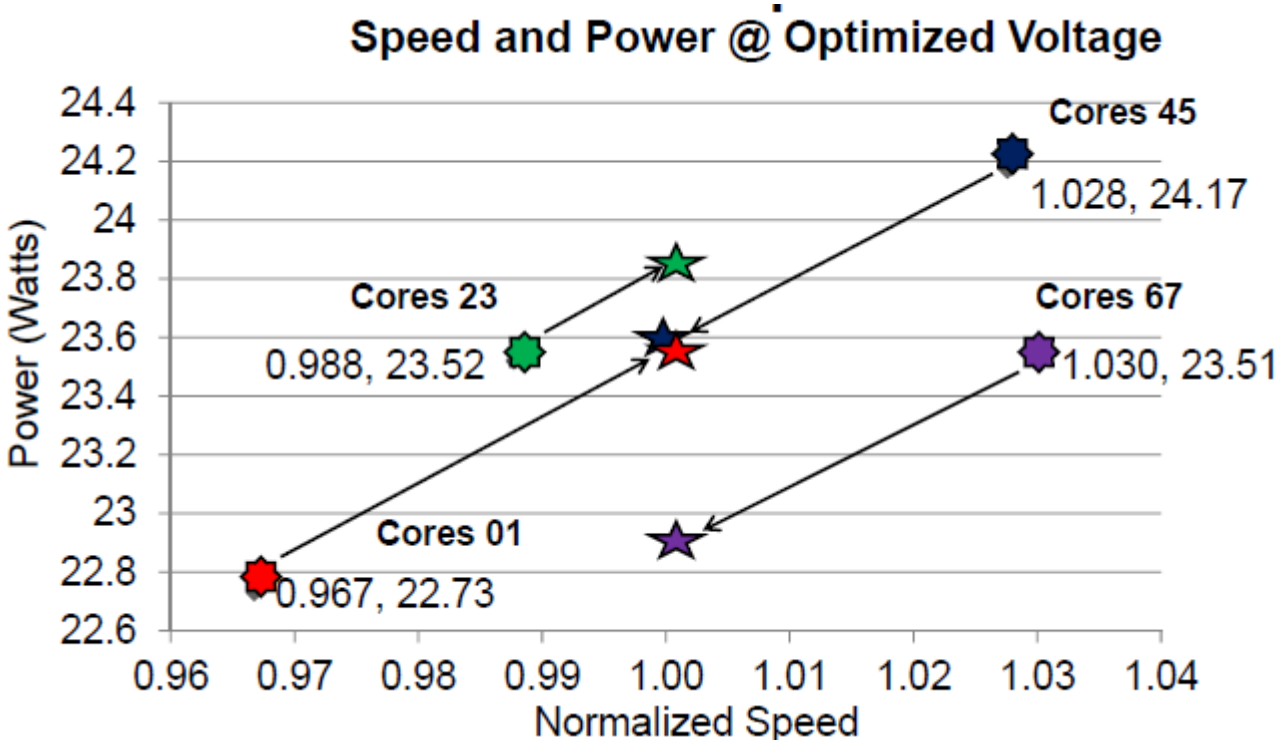
Speed and Power can be optimized for each Voltage domain for improved performance

Core pair optimization-2 [54]



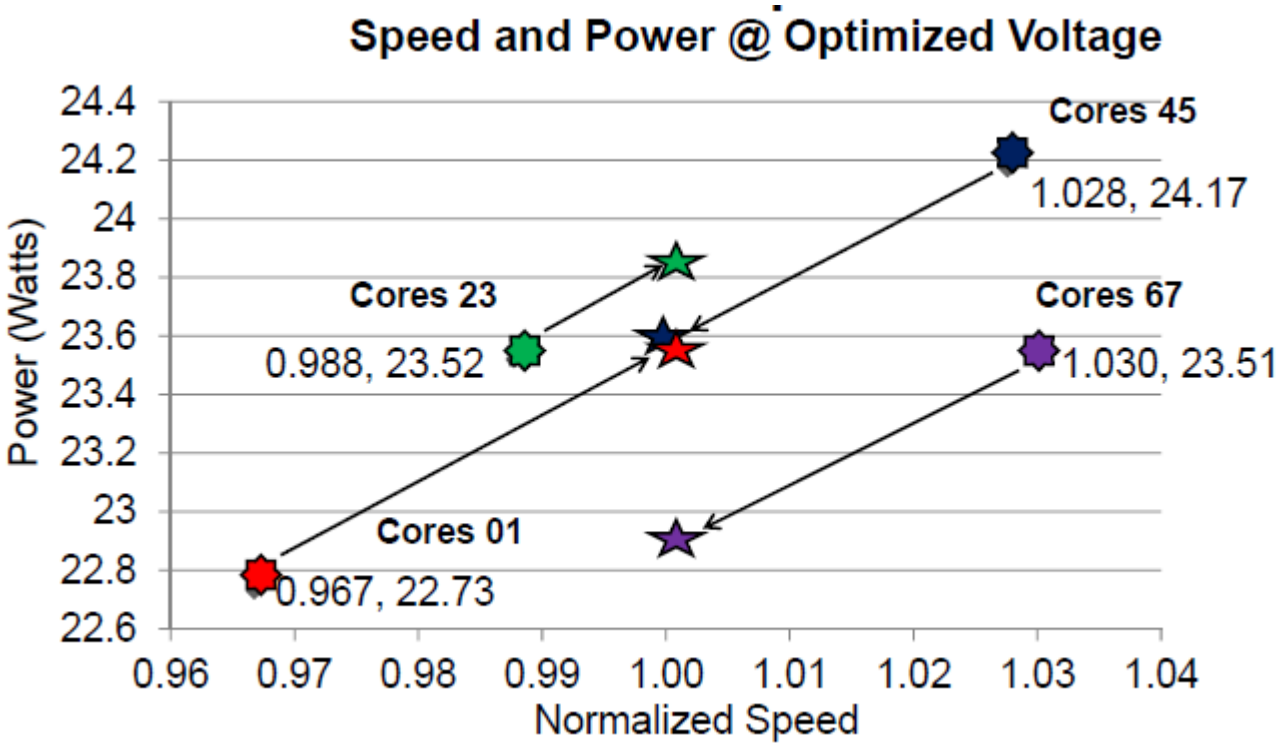
- Cores 01 are slower and lower power
  - Increase voltage and power to improve frequency
- Cores 45 are faster and higher power
  - Decrease voltage and frequency to recover power

Core pair optimization-3 [54]



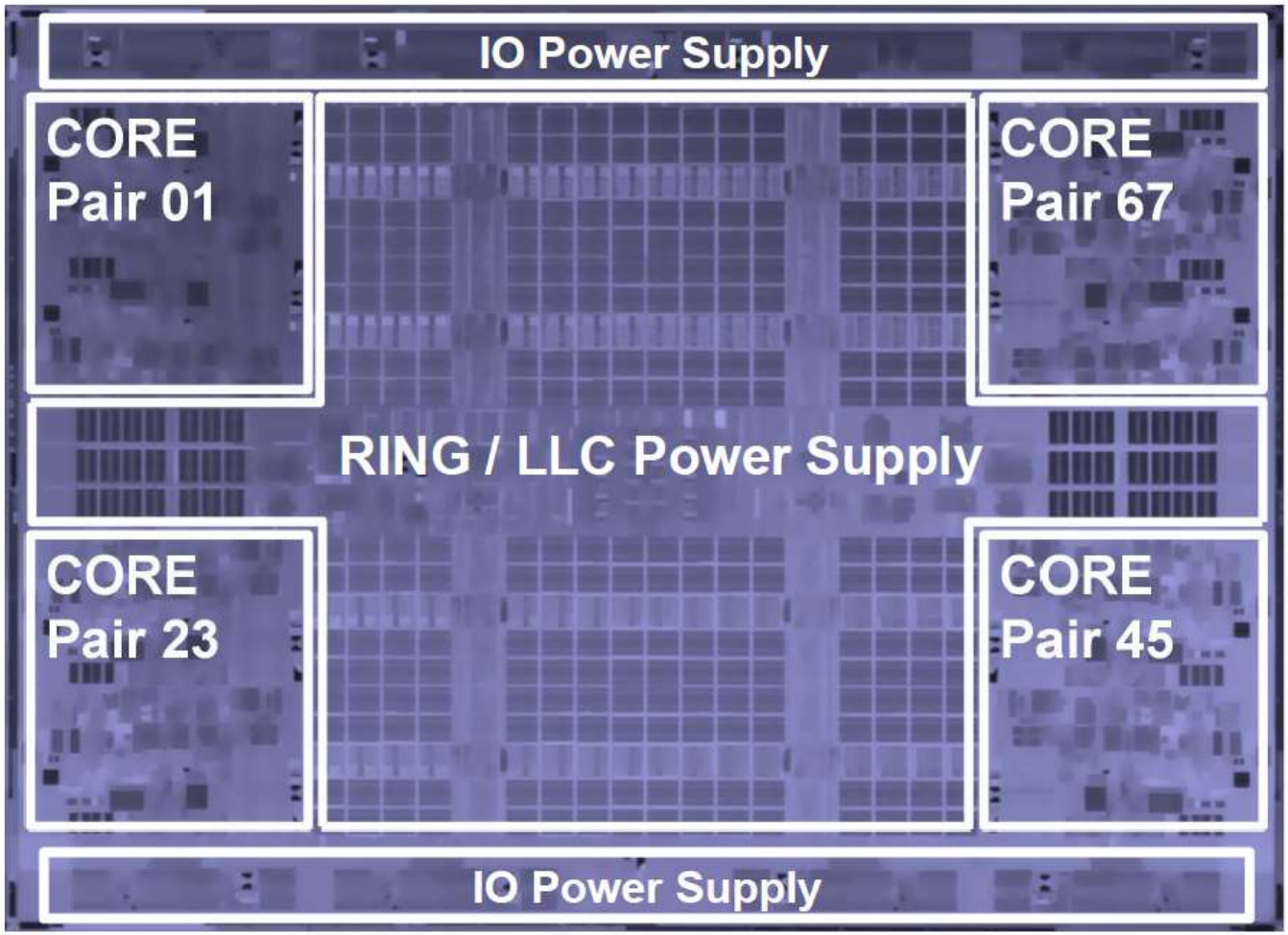
- Cores 23 are slower and lower power
  - Increase voltage and power to improve frequency
- Cores 67 are fast and low power
  - Decrease voltage and frequency to recover power

Core pair optimization-4 [54]

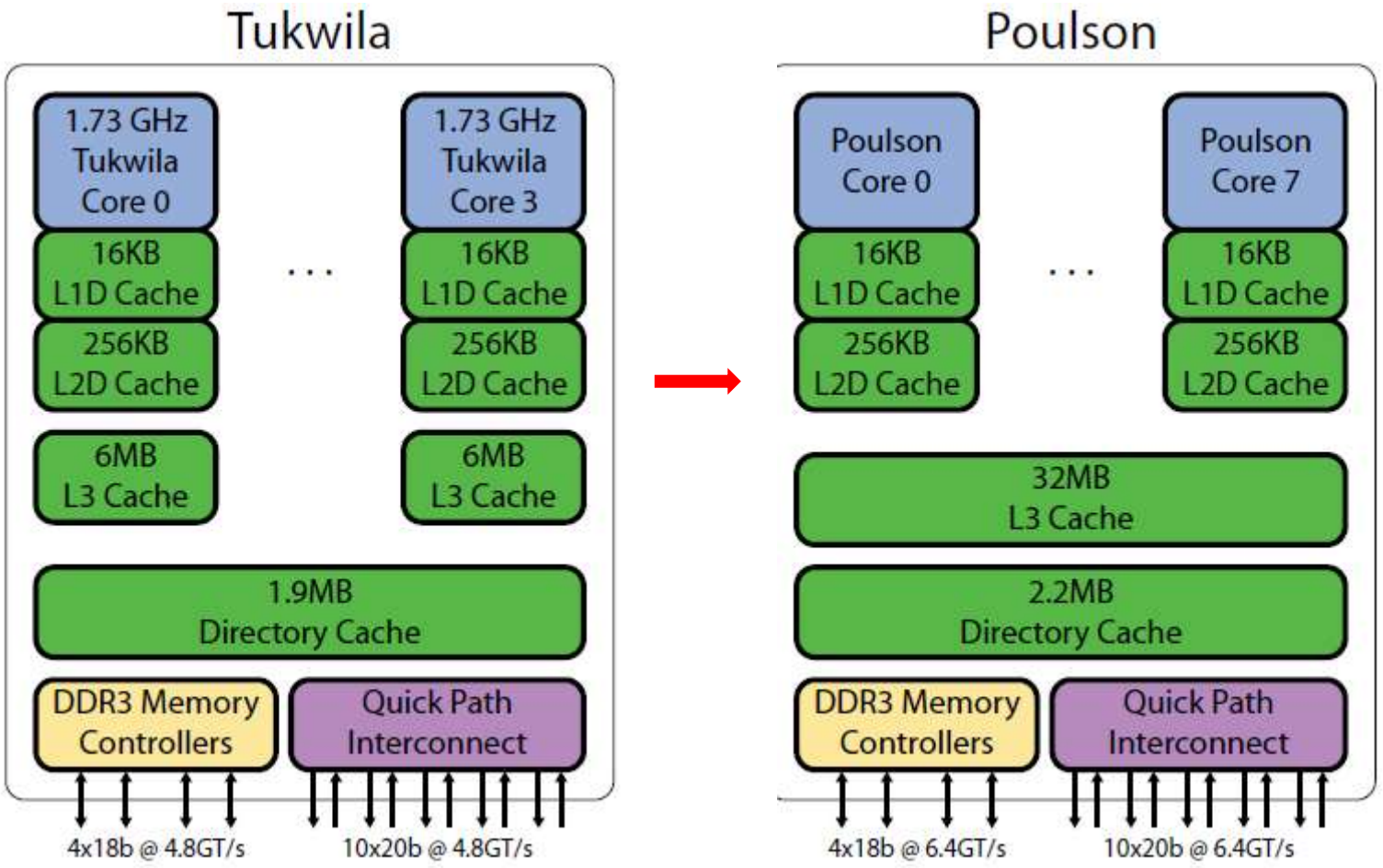


Independent supply optimization improves frequency up to 5% with no impact to power!

Processor power planes [54]

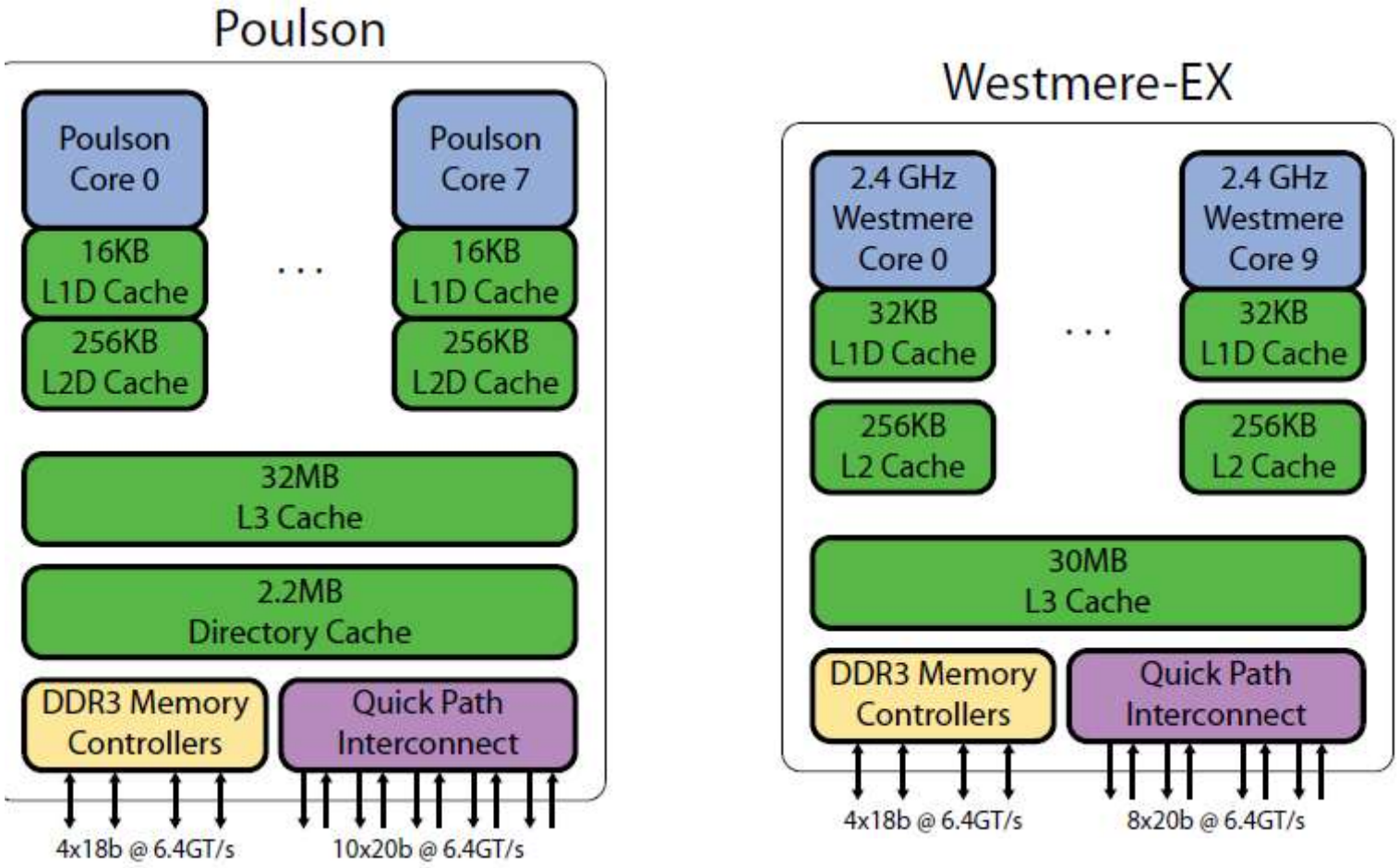


d3) Enhanced cache architecture of Poulson [31]



7.3 The 8-core Poulson (Itanium 9500) line (23)

Contrasting the cache architecture of Poulson to the Westmere-EX design [31



d5) Increased QPI speed

The transfer speed of Poulson's could be increased from 4.8 GT/s (as implemented in the Tukwila) to 6.4 GT/s.

d5) RAS improvements [58]

	Itanium 9300	Poulson
LLC Data	SECDEC	DECTED
LLC Tags	SECDEC	SECDED
MLI Data & Tags	Parity w/inval	SECDED
MLD Data & Tags	SECDED	SECDED
Directory Cache	Parity w/inval	SECDED
Integer/Floating Point Registers	Parity	SECDED
IBD	None	Parity w/replay
FPU Adders and Multipliers	None	Residual
End-to-End Data Detection	No	Yes
Memory Controller	x4=DDDC, x8=SDDC	x4=DDDC, x8=SDDC
Memory Sparing	DIMM	Rank
SMI lane/clock failover	Reduced ECC	Full ECC

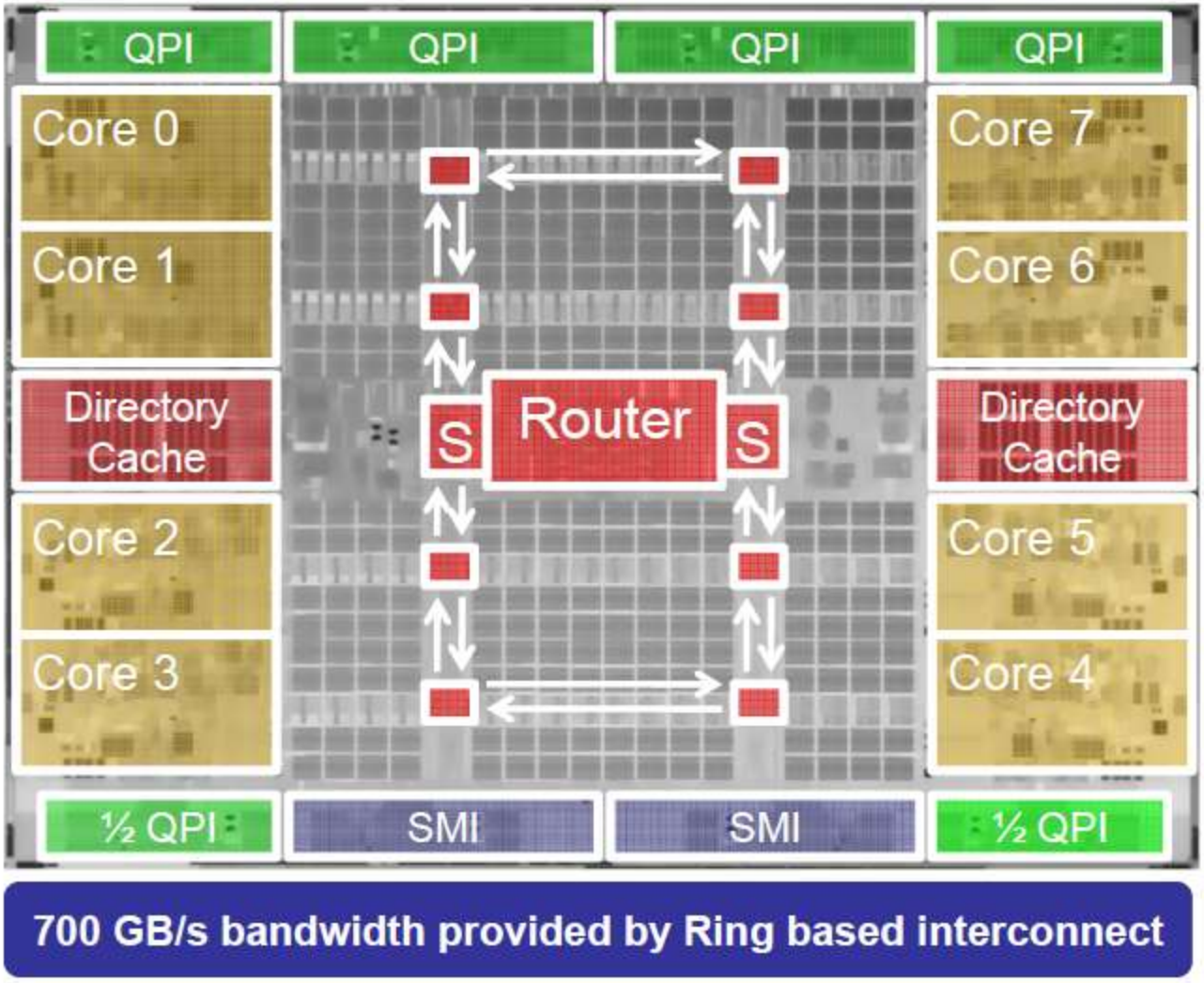
e) On-die ring interconnect

Before the Poulson line Intel already introduced on-die ring interconnect into a number of processor designs, including

- Nehalem-EX server line (3/2010)
- Sandy Bridge mobile and desktop lines (1/2011)
- Westmere-EX line (4/2011).

The ring interconnect of Poulson consists of two counter rotating rings, as shown in the next slide, protected by parity bits [59].

On-die ring interconnect of the Poulson line [54]



7.3 The 8-core Poulson (Itanium 9500) line (28)

Remarks to Poulson's microarchitecture [31], [60]

1) Poulson makes further on use of verniers introduced in Tukwila, that is delay programmable buffers in the last stage of the clock that can shift the clock edge forwards or backwards, as needed.

   Poulson's verniers have a delay range of roughly 30ps, which is about a sixteenth of a full clock cycle.

   There are altogether about 45 000 clock skew adjustment points incPoulson's die. According to industry sources, the verniers technology raises Poulson's clock speed by about 400 MHz.

2) Similarly to Tukwila, Poulson provides also on-die directory supported cache coherency, as indicated in the next Figure.

   There are two directories each of 1.1 or 1.5 MB size on the die (there are two different figures in Intel's publications).

Directory caches on Poulson's die [54]

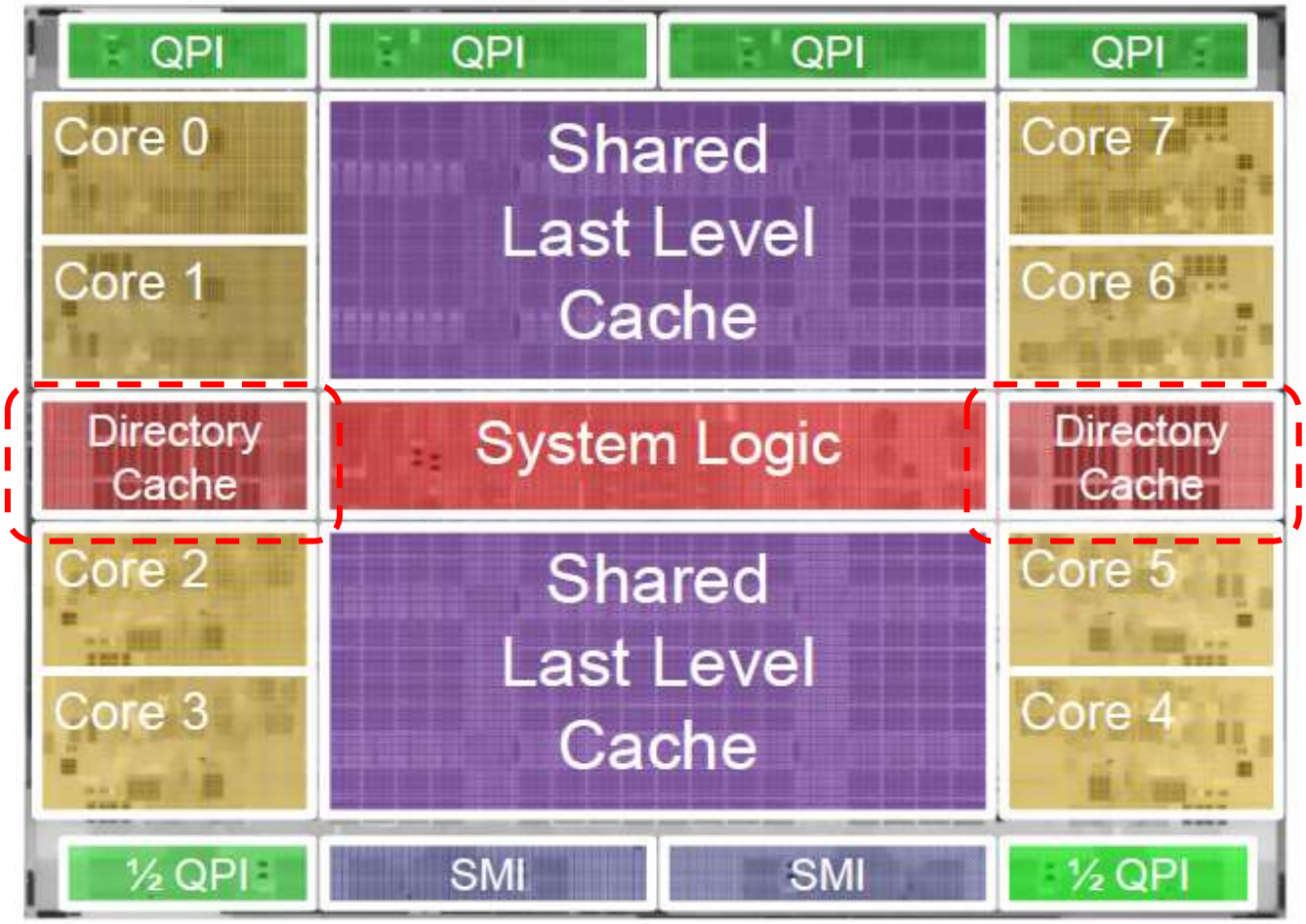
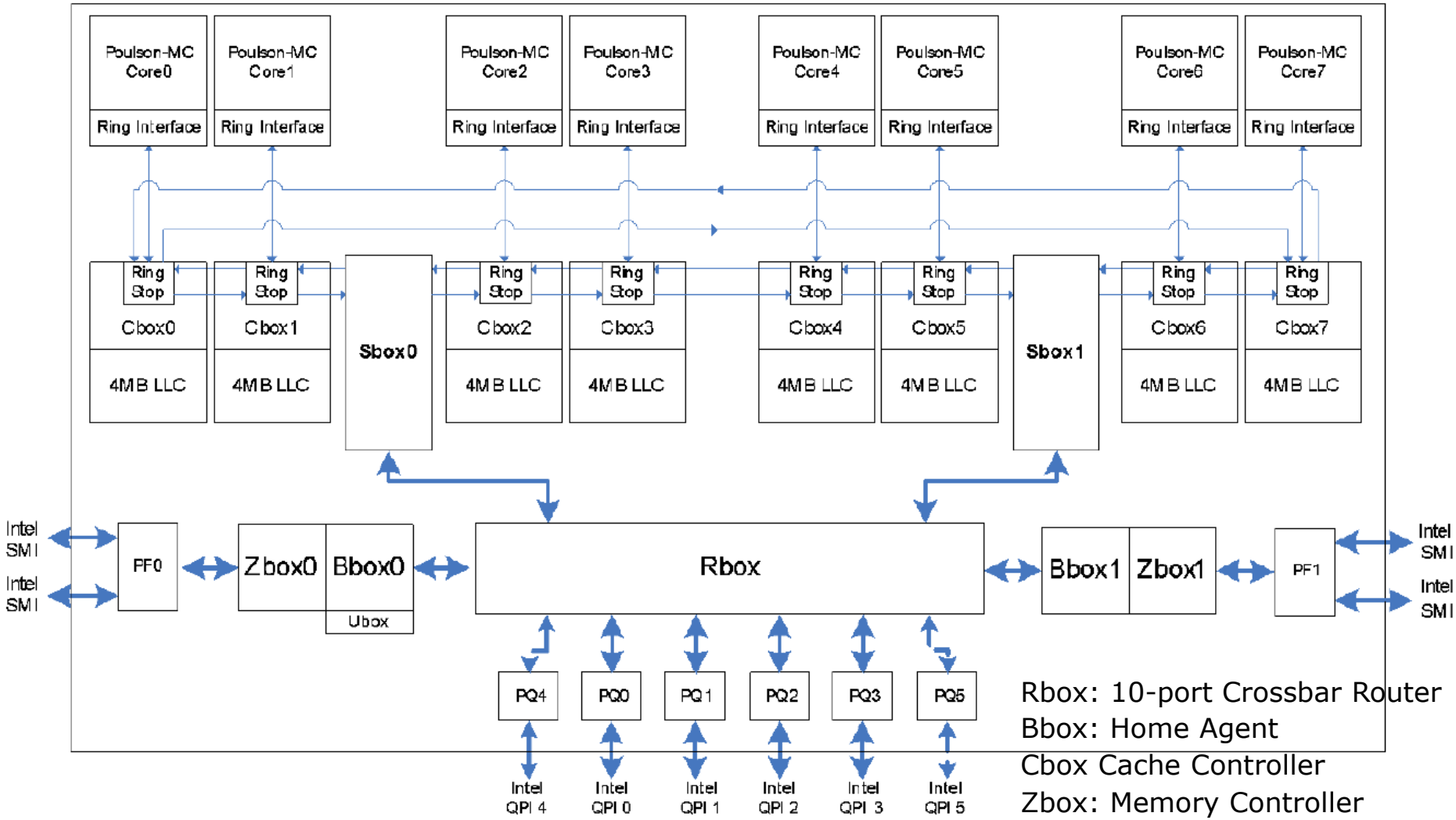


Figure: Block diagram of the Poulson processors [54]

7.3 The 8-core Poulson (Itanium 9500) line (29b)

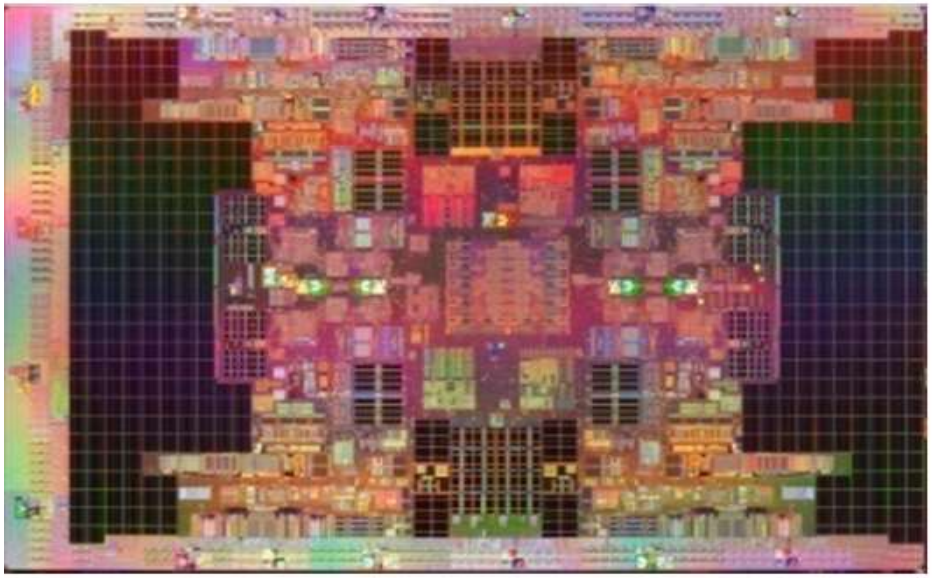
Intel Itanium 9500 and 9700 series block diagram [71]



7.3 The 8-core Poulson (Itanium 9500) line (30)

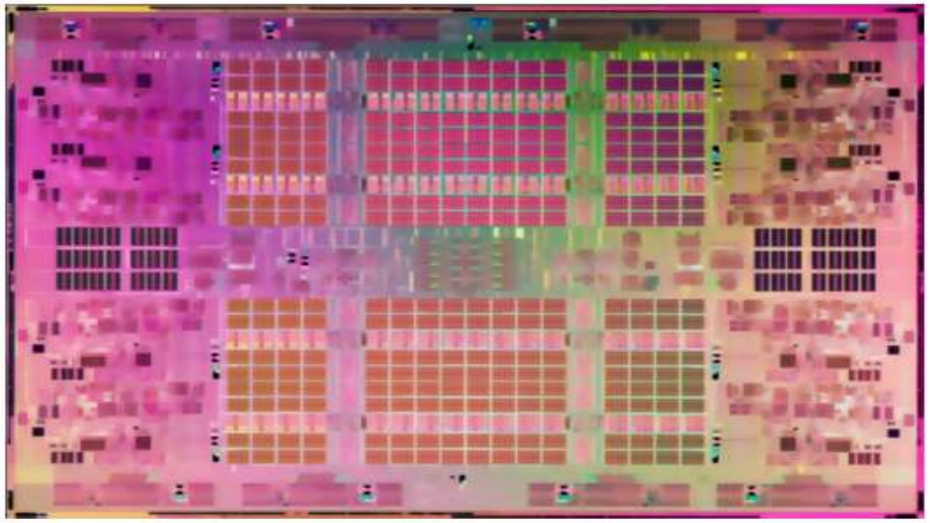
New chip layout with outside cores [54]

Tukwila



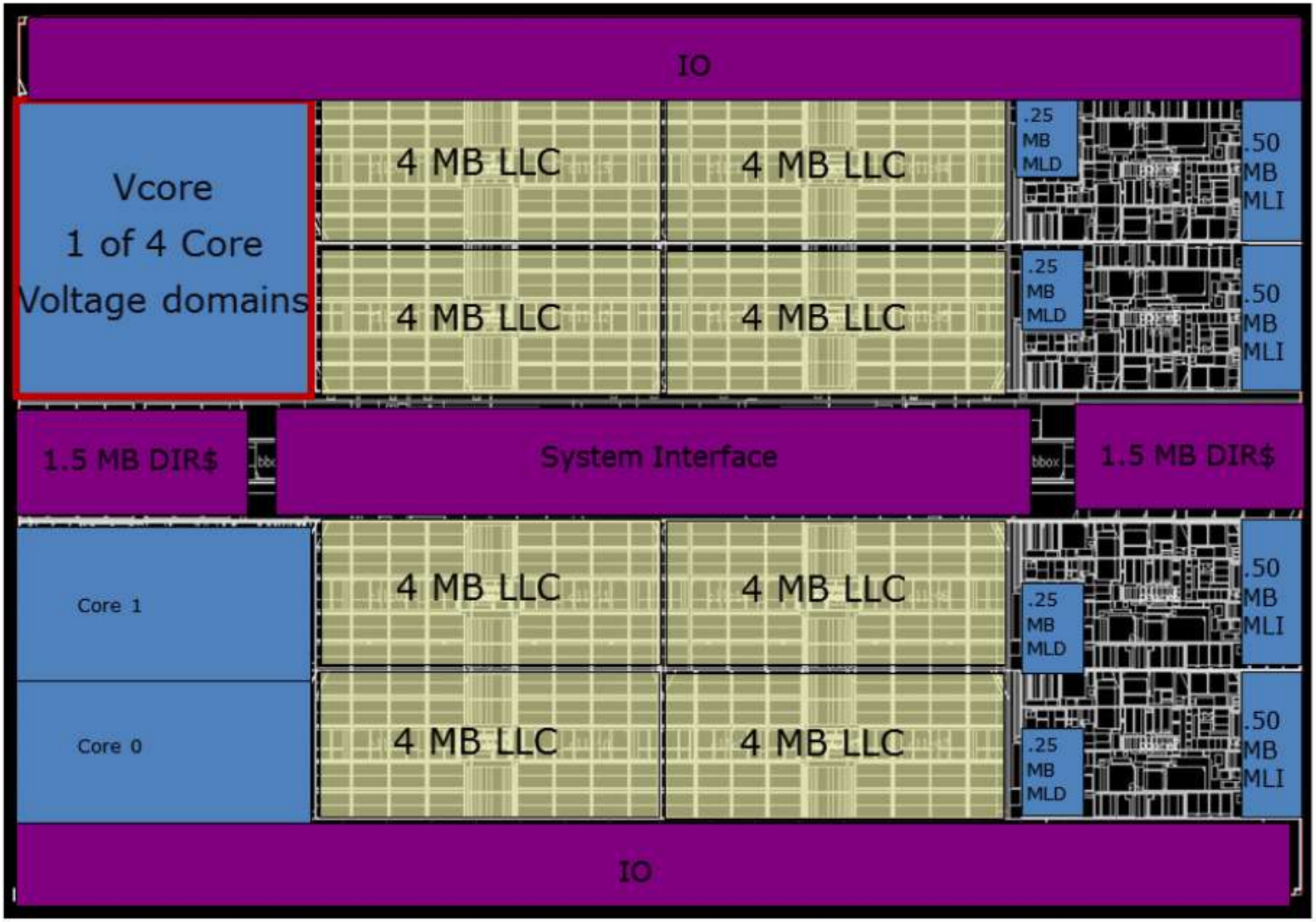
65 nm
699 mm^2
2.05 billion transistors

Poulson



32 nm
544 mm^2
3.1 billion transistors

Poulson's floor plan [60]



Contrasting the chip statistics of Poulson and Tukwila [31]

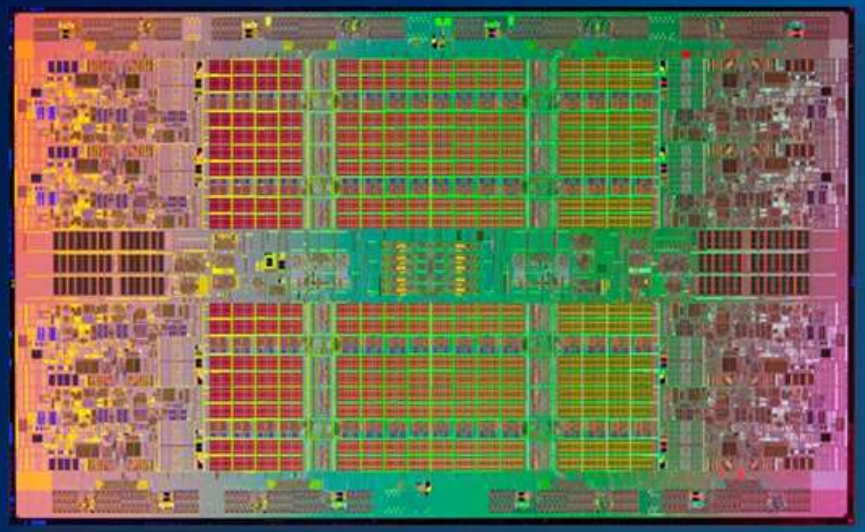
	Poulson				Tukwila			
	Area	Devices	Voltage	Power*	Area	Devices	Voltage	Power**
	mm^2	M	V	W	mm^2	M	V	W
Cores	158	712	0.85-1.2	95	276	430	0.9-1.2	95
L3 Cache	163	2173	0.9-1.1	5	191	1420	1.1	40
System	137	224	0.9-1.1	50	107	152	0.9-1.2	30
I/O	68	44	1.05-1.1	20	123	39	1.1	20
Other	18				3	4		
Total	**544**	**3153**		**170**	**700**	**2045**		**185**

*Intel estimates

**Estimates, based on 185W TDP and Itanium 9300 electrical specification

Key advancements of Poulson over Tukwila [57]

Most Sophisticated Intel Processor To Date



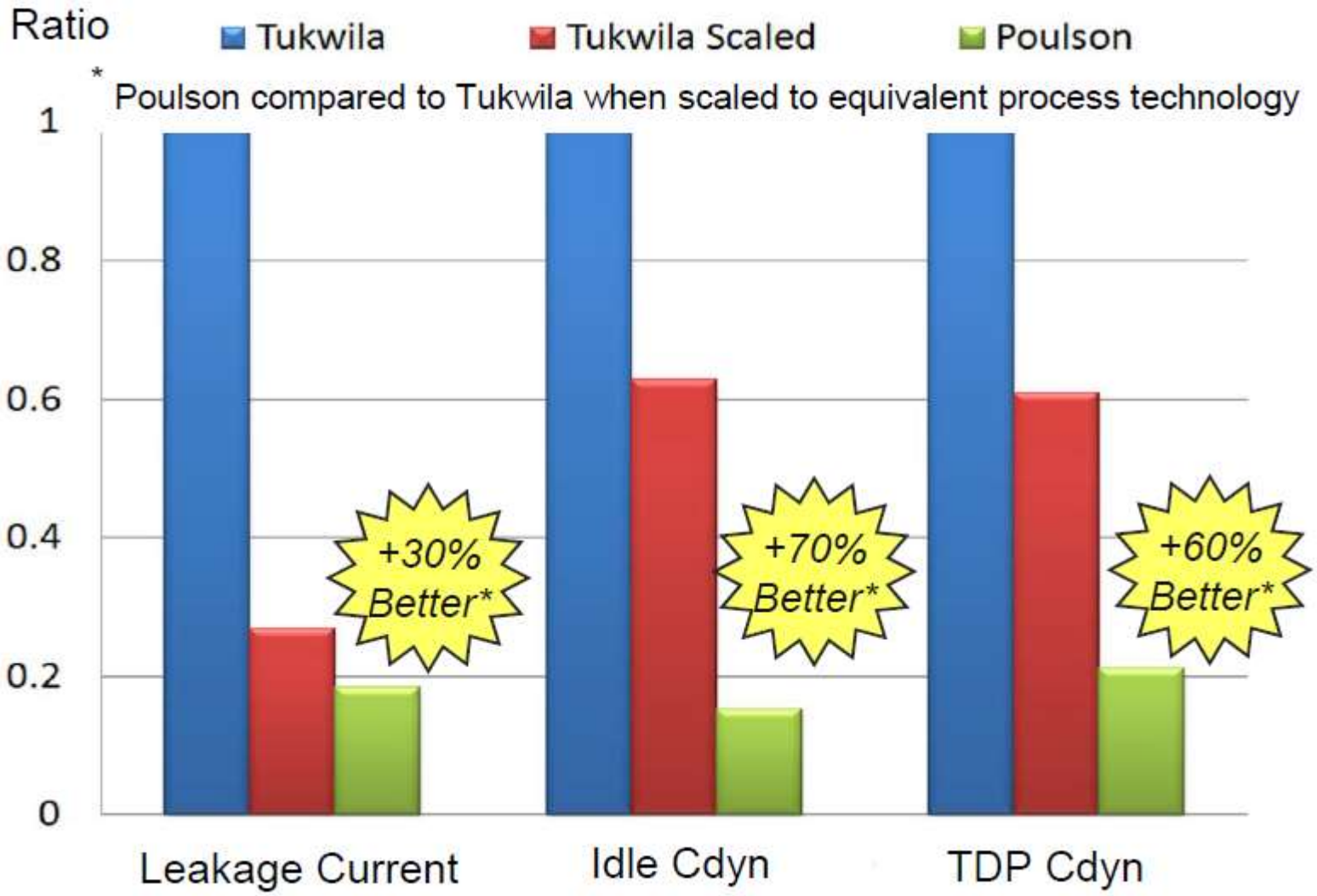
Advancements over Itanium 9300

- New microarchitecture design
- 2x the cores[1], 2x instruction throughput[2]
- Up to 2.53Ghz frequency
- Up to 8% lower TDP[3], 80% reduced idle power[4]

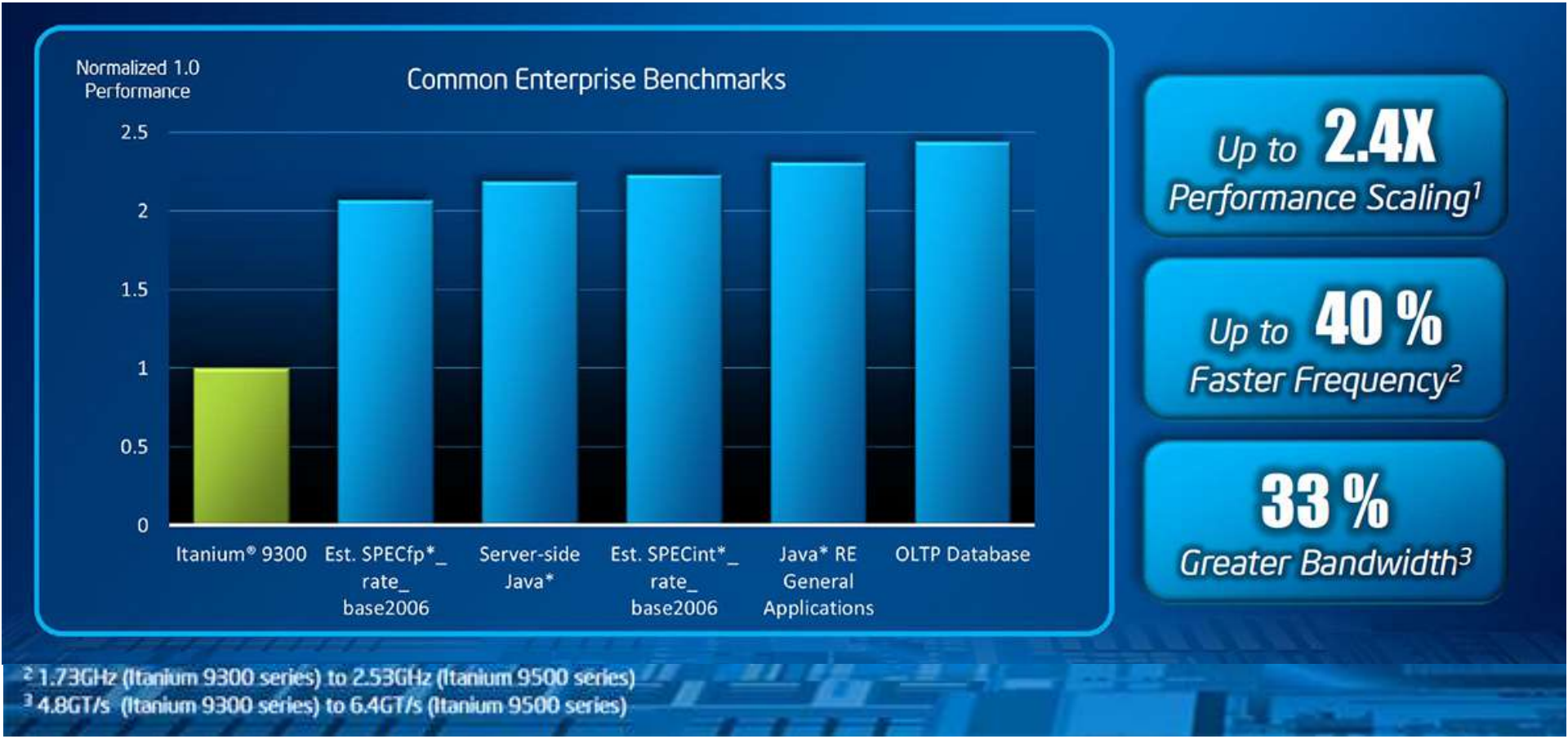
[1] 4 cores to 8 cores (Itanium 9300 series vs. Itanium 9500 series)
[2] Maximum 6-wide vs. 12-wide instruction issue (Itanium 9300 series vs. Itanium 9500 series)
[3] 185W TDP vs. 170W TDP (Itanium 9300 series vs. Itanium 9500 series)
[4] Source: Intel internal measurements comparing individual core Idle dynamic power (Itanium 9500 series vs. itanium 9300 series)

7.3 The 8-core Poulson (Itanium 9500) line (34)

The effect of technology scaling (from 65 nm to 32 nm) on the dissipation [5

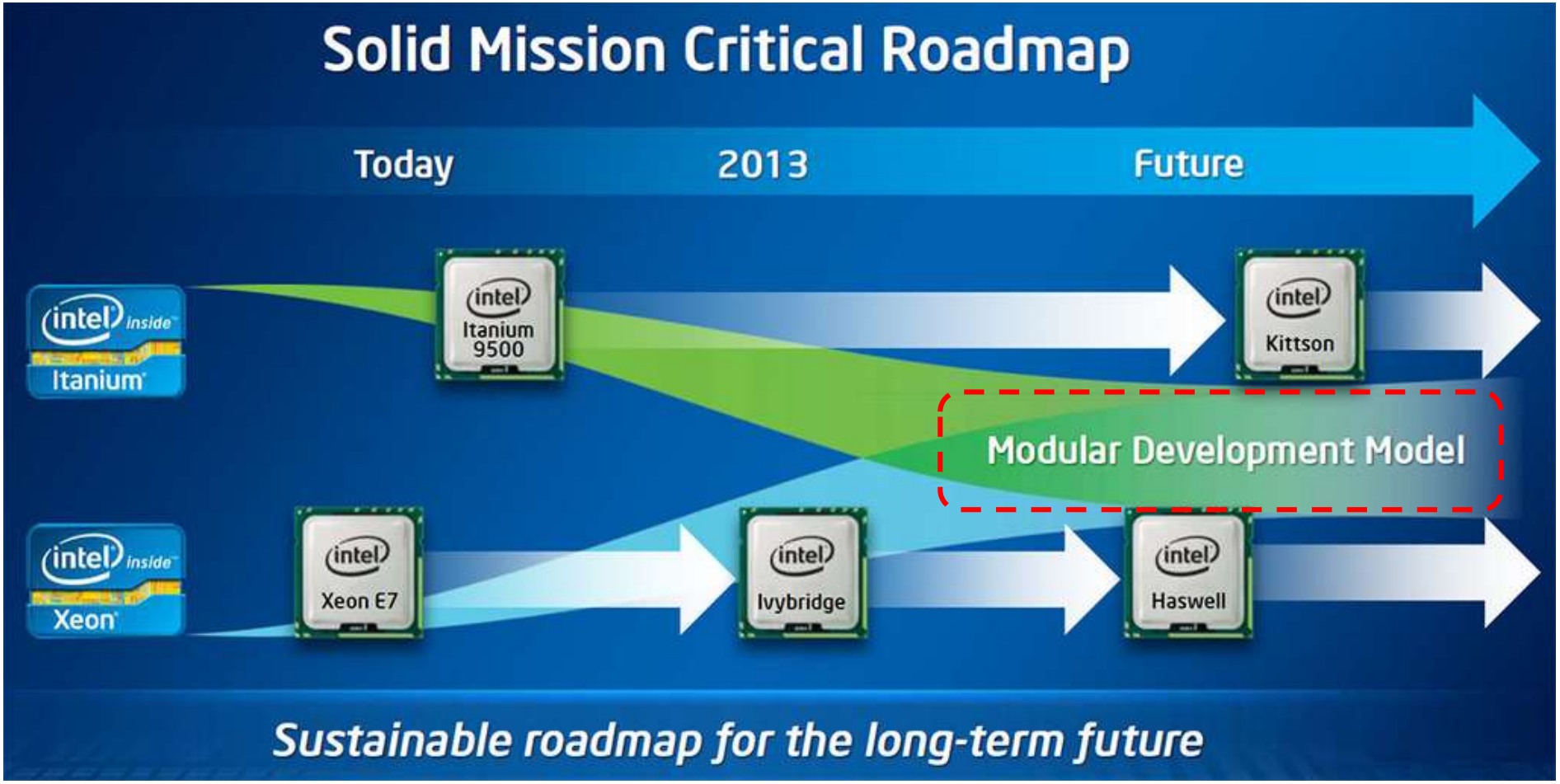


Simulation-based performance estimates of Poulson [57]



7.4 The Kittson line

Intel's mission critical roadmap from 11/2012 [61]



7.4 The Kittson line-2 [61]

During the introduction of the Poulson processor (11/2012) Intel announced that future generations of Itanium processors (practically Kittson) will adopt an innovative "Modular Development Model", as seen in the next Figure.

7.4 The Kittson line (3)

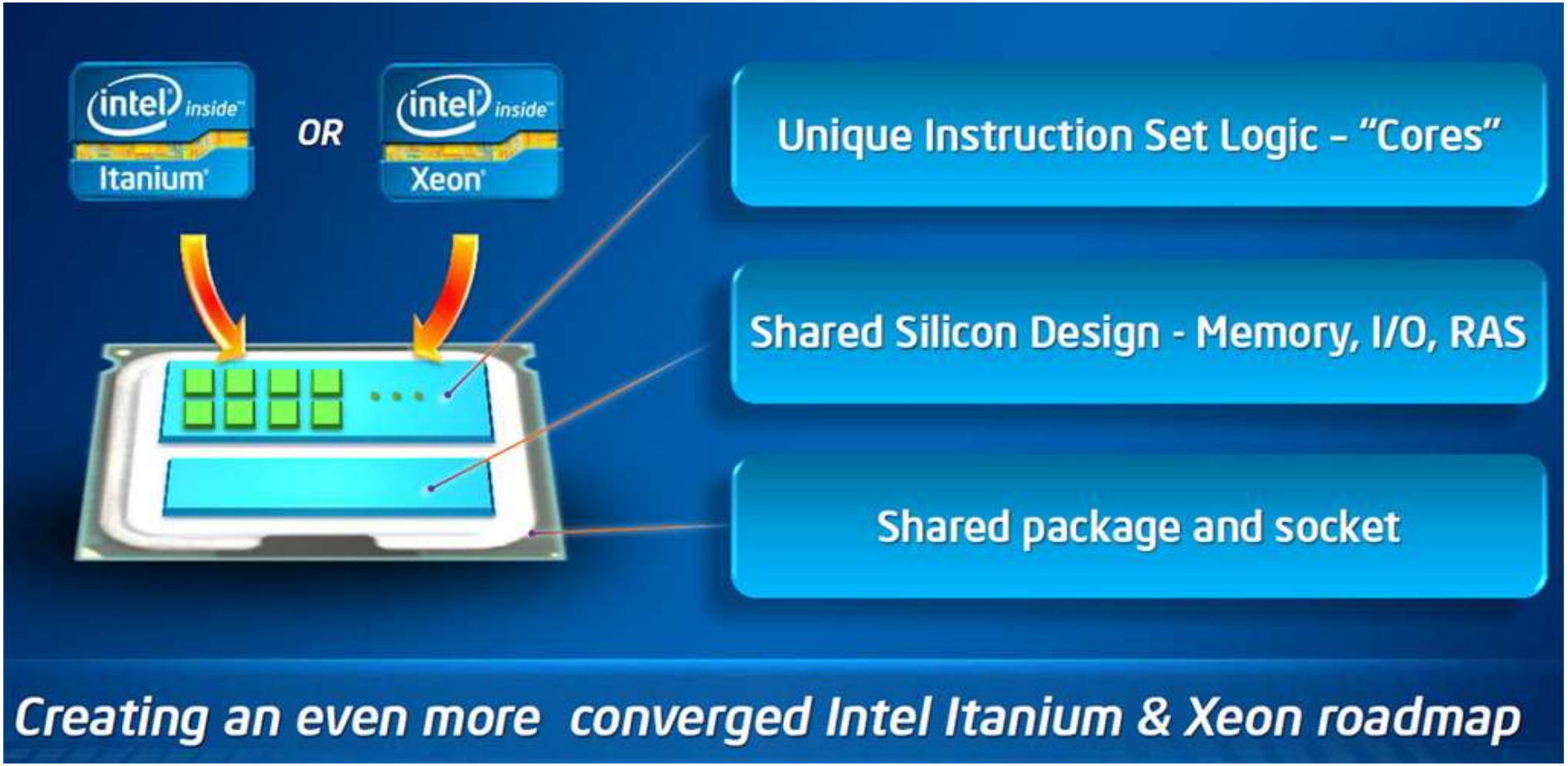
The modular development model

It will

- retain unique ISAs (IA-32/IA-64 cores) but will have
- shared silicon design (memory, I/O, RAS) and
- shared package and socket,.

as indicated in the next Figure.

Intel's "Modular Development Model" [61]



7.4 The Kittson line (5)

Modification of Intel's plan concerning the Kittson processor

Nevertheless, in 1/2013 Intel announced that they modified their roadmap and Kittson would not be shrunken to 22 nm (as originally planned) but will be manufactured on Intel's 32 nm process technology and will remain socket compatible with the existing 9300/9500 platform.

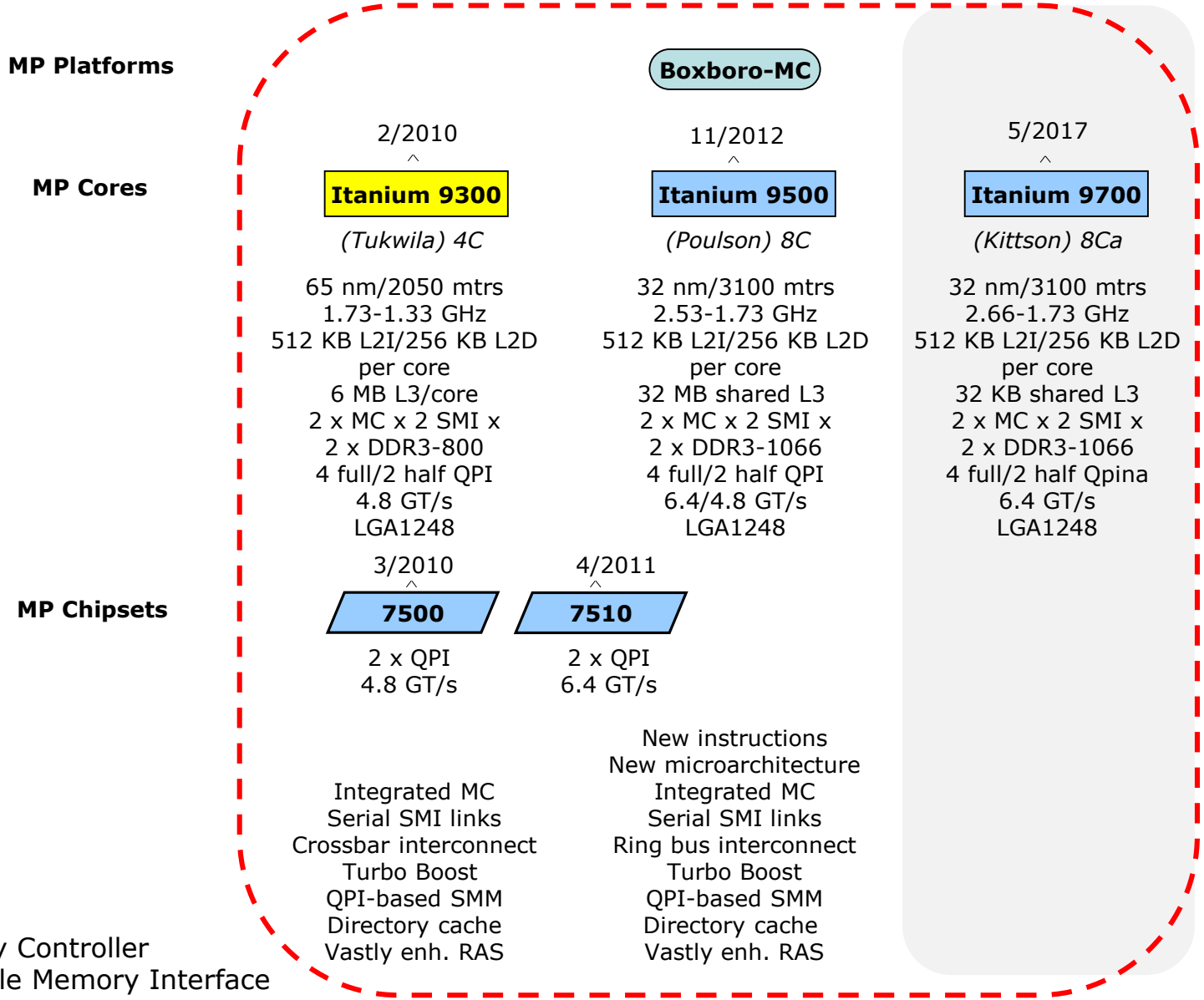
This means first that no major advancements can be expected from the Kittson processor and second that Intel withdraw their plan to work on a Modular Development Model.

7.4 The Kittson line (6)

The Kittson line as introduced finally in 5/2017 [69]

The Kittson line has no additional features or capabilities vs. the previous 9500 line (Poulson) of processors, only it provides slightly higher clock speeds for certain processor models.

7.4 The Kittson line compared to the previous lines



MC: Memory Controller
SMI: Scalable Memory Interface

Main features of the Itanium 9700 (Kittson) server line [70]

Intel Itanium (Kittson) CPUs					
	Cores/ Threads	Base Freq	L3	TDP	Cost*
Itanium 9760	8/16	2.66 GHz	32 MB	170 W	$4650
Itanium 9750	4/8	2.53 GHz	32MB	170W	$3750
Itanium 9740	8/16	2.13 GHz	24 MB	170 W	$2650
Itanium 9720	4/8	1.73 GHz	20 MB	130 W	$1350

8. The end of life of the Itanium family

8. The end of life of the Itanium family-1

About 1990 Intel presented the Itanium line as the next generation architecture for the next 25+ years, as indicated below.

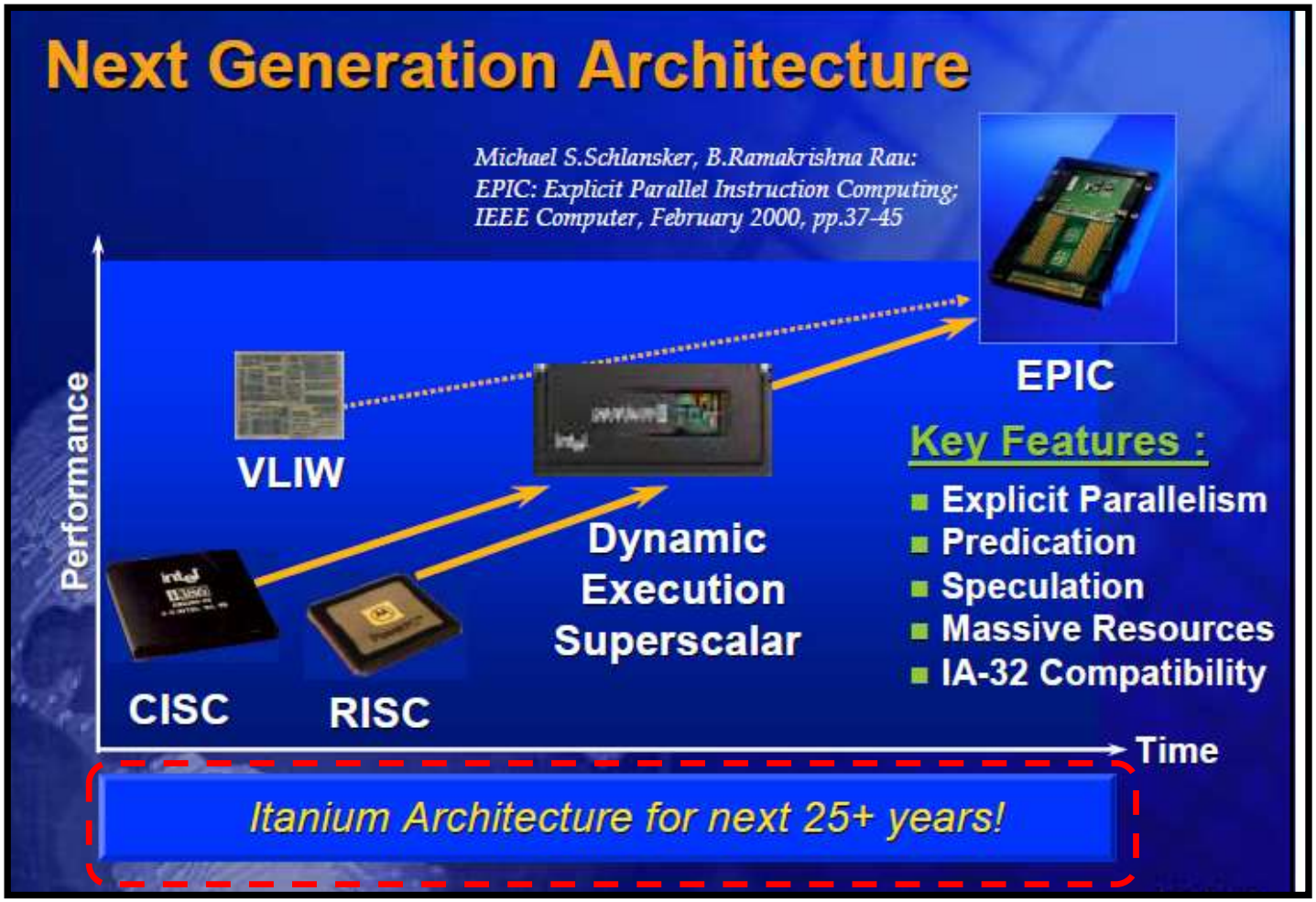


Figure: Introducing the Itanium line about 1990 [11]

8. The end of life of the Itanium family (2)

The end of life of the Itanium family-2 [62], [63], [64]

Provoked by the shrinking market interest for the Itanium family and more or less in concert with its projected lifespan, in the beginning of the 2010's major software vendors gradually discontinued its support as follows:

- In 1/2010 Red hat announced that they end the support of the Itanium line with the next release of their Linux distribution.
- In 4/2010 Microsoft disclosed that Windows Server R2 will be the last OS server version to support the Itanium architecture.
- Also the SQL Server 2008 R2 and Visual Studio 2010 are the last versions to support Itanium.
- In 3/2011 also Oracle has decided to discontinue all software development on the Itanium microprocessor stating that Intel management made it clear that their strategic focus is on their x86 microprocessor and that Itanium was nearing the end of its life.

  Nevertheless, HP who sells Itanium processors, fought Oracle decision referring to a related contract.

  The court backed HP's claim and in 9/2012 Oracle announced that they will continue to build software for the Itanium family.

8. The end of life of the Itanium family (3)

The end of life of the Itanium family-3 [65]

In 1/2013 Intel announced that they modified their roadmap and Kittson would not be shrunken to 22 nm but will be manufactured on Intel's 32 nm process technology and will be socket compatible with the existing 9300/9500 platform.

This means that no major advancements can be expected from Kittson and probably Kittson will be the last member of the IA-64 family.

Finally, in 05/2017 Intel launched their 9700 Series (Kittson) Itanium line as the last line of the Itanium family.

9. References

References (1)

[1]: Kerner M., Padgett N., A History of Modern 64-bit Computing, Febr. 2007, http://courses.cs.washington.edu/courses/csep590/06au/projects/history-64-bit.pdf

[2]: Smotherman M., Historical background for EPIC, Nov. 2012, http://people.cs.clemson.edu/~mark/epic.html

[3]: Smotherman M., Understanding EPIC Architectures and Implementations, http://people.cs.clemson.edu/~mark/464/acmse_epic.pdf

[4]: Crawford J., Introducing the Itanium Processors, IEEE Micro, Sept.-Oct. 2000, http://eecs.vanderbilt.edu/courses/eece343/papers/IA64-Micro-Sep00.pdf

[5]: Kanellos M., Crothers B., Intel, HP unveil EPIC technology, CNet, Oct. 14 1997, http://news.cnet.com/Intel,-HP-unveil-EPIC-technology/2100-1001_3-204260.html

[6]: Kanellos M., Intel names Merced chip Itanium, CNet, Oct. 4 1999, http://archive.is/IUNih#selection-1095.1-1095.222

[7]: DeMone P., Countdown to IA-64, Real World Technologies, March 14 2001, http://www.realworldtech.com/countdown-to-ia64/

[8]: Shankland S., Itanium: A cautionary tale, ZDNet, Dec. 7 2005, http://web.archive.org/web/20080209211056/http://news.zdnet.com/2100-9584-5984747.html

[9]: Gwennap L., Intel's Itanium and IA-64: Technology and Market Forecast, MDR, 2000

[10]: Lancaster G.L., Changing the Economics of Computing Thru Technology Leadership, IBM Forum 2005, http://www.dbguide.net/upload/20050411/111320965 2764.pdf

References (2)

[11]: Kilian D., Intel Itanium Processor Family, Technical Marketing, Intel GmbH, Munich, April 18 2002, http://www.decus.de/slides/sy2002/18_04/3e02.pdf

[12]: Schlansker M.S., Rau B.R., EPIC: Explicitly Parallel Instruction Computing, Hewlett-Packard Laboratories, Computer, Febr. 2000, https://courses.cs.washington.edu/courses/cse471/01au/epic_cgi.pdf

[13]: Cornelius H., Intel Itanium Architecture, Intel EMEA, Munich, Jan. 28 2003, https://www.rrze.uni-erlangen.de/dienste/arbeiten-rechnen/hpc/vortraege/IntelCornelius.pdf

[14]: Sharangpani H., Intel Itanium Processor Core, Hot Chips, Aug. 15 2000, http://www.hotchips.org/wp-content/uploads/hc_archives/hc12/hc12pres_pdf/itanium-core.pdf

[15]: Wall D.W., Limits of Instruction-Level Parallelism, WRL Technical Note TN-15, Dec. 1990, http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-15.pdf

[16]: Kilian D., Itanium Processor Family Architecture, A High-Performance Computing Architecture Dec. 2000, http://www.decus.de/slides/sy2002/18_04/3e02.pdf

[17]: Doshi G., Understanding the IA-64 Architecture, Aug. 31 – Sept. 2 1999, http://www.iuma.ulpgc.es/~nunez/procesadoresILP/intel-idf-isa64.pdf

[18]: Huck J., Morris D., Ross J., Knies A., Mulder H., Zahir R., Introducing the IA-64 Architecture, IEEE Micro, Sept-Oct. 2000, http://www.cs.ucr.edu/~bhuyan/cs162/LECTURE7a.pdf

[19]: Lamberts S., IA-64 Architecture, Overview, A High-Performance Computing Architecture, July 2000, http://www.cs.helsinki.fi/u/kerola/tikra/IA64-Architecture.pdf

References (3)

[20]: Wikipedia, Advanced load address table, http://en.wikipedia.org/wiki/Advanced_load_address_table

[21]: Wikipedia, Register window, http://en.wikipedia.org/wiki/Register_window

[22]: HP/Intel IA-64, http://www.seas.gwu.edu/~narahari/cs211/materials/lectures/IA-64.pdf

[23]: Sima D., Fountain T., Kacsuk P., Advanced Computer Architectures: A Design Space Approach Addison-Wesley, 1997

[24]: Sharangpani H., Arora K., Itanium Processor Microarchitecture, IEEE Micro, Sept-Oct. 2000, http://users.ece.gatech.edu/~leehs/ECE6100/papers/ia64uarch.pdf

[25]: Wikimedia, File:Itanium arch.png, http://commons.wikimedia.org/wiki/File:Itanium_arch.png

[26]: Orlowski A., Benchmarks – Itanic 32bit emulation is 'unusable', The Register, Jan. 23 2001, http://www.theregister.co.uk/2001/01/23/benchmarks_itanic_32bit_emulation/

[27]: Baraz L., Devor T., Etzion O. et al., IA-32 Execution Layer: a two-phase dynamic translator designed to support IA-32 applications on Itanium®-based systems, http://software.intel.com/sites/default/files/m/d/4/1/d/8/93085_93085.pdf

[28]: Alex I., Part I: IA-32 Execution Layer, Part II: 64-bit Extension Technology

[29]: Samaras B., The Itanium Processor Cartridge, Hot Chips, 2000, http://www.hotchips.org/wp-content/uploads/hc_archives/hc12/hc12pres_pdf/itanium-cartridge.PDF

References (4)

[30]: Intel 460GX Chipset System Software Developer's Manual, June 2001, http://data.manualslib.com/pdf/8/777/77665-intel/460gx.pdf?4e8bdc3ba3cbd614594383c3f9b5bd53&take=binary

[31]: Kanter D., Poulson: The Future of Itanium Servers, Real World Technologies, May 18 2011, http://www.realworldtech.com/poulson/3/

[32]: Intel Itanium 2 Processor Reference Manual, For Software Development and Optimization, May 2004, http://people.freebsd.org/~marcel/refs/ia64/itanium2/25111003.pdf

[33]: Bhandarkar D., Golliver R., The Road to Billion Transistor Processor Chips in the Near Future, April 22 2003, http://www.lanl.gov/orgs/hpc/salishan/salishan2003/golliver.pdf

[34]: Intel Itanium 2 1.7GHz Socket PAC611 CPU 9MB L3 Cache Q84Z, Ebay, http://www.ebay.com/itm/Intel-Itanium-2-1-7GHz-Socket-PAC611-CPU-9MB-L3-Cache-Q84Z-/330709240444

[35]: Azimi M., Briggs F., Cekleov M., Khare M., Kumar A., Looi L.P., Scalability Port: A Coherent Interface for Shared Memory Multiprocessors, IEEE Xplore, 2002

[36]: Grudinski A., Itanium 2 Platform and Technologies, 2004, http://de.openvms.org/TUD2005/08_IPF_Directions__Alexander_Grudinski_Intel.pdf

[37]: Dual-Core Update to the Intel Itanium 2 Processor Reference Manual, Rev. 0.9, Jan. 2006, http://www.intel.com/content/www/us/en/processors/itanium/dual-core-update-itanium-2-processor-manual.html

References (5)

[38]: Naffziger S., Stackhouse B., Grutkowski T., The Implementation of a 2-core, Multi-threaded Itanium Family Processor, 2005, http://ewh.ieee.org/r5/denver/sscs/Presentations/2005_03_Montecito1.pdf

[39]: Mahoney P., Fetzer E., Doyle B., Naffziger S., Clock Distribution on a Dual-core Multi-threaded Itanium Family Microprocessor, 2005, http://ewh.ieee.org/r5/denver/sscs/Presentations/2005_03_Montecito2.pdf

[40]: Kubicki K., The Count of "Montecito", Daily Tech, July 16 2006, http://www.dailytech.com/article.aspx?newsid=2907

[41]: Dr. Liu F., Intel Itanium Architecture Update, Oct. 5 2006, http://de.openvms.org/TUD2006/03_Intel_Intanium_Update.pdf

[42]: Ignowski J., McGowen R., Poirier C., Bostak C., Naffziger S., Power and Temperature Control on a 90nm Itanium Family Processor, 2005, http://ewh.ieee.org/r5/denver/sscs/Presentations/2005_03_Montecito4.pdf

[43]: Morgan T.P., Intel Pushes Out Itaniums, Replaces Future Xeon MPs, IT Jungle, Oct. 26 2005, http://www.itjungle.com/two/two102605-story01.html

[44]: Needle D., AMD Snares Itanium Designer, Internet News, March 30 2006, http://www.internetnews.com/bus-news/article.php/3595561/AMD+Snares+Itanium+Designer.htm

[45]: Huynh A.T., Intel Readies Itanium 2 9100-Series, Daily Tech, July 20 2007, http://www.dailytech.com/Intel+Readies+Itanium+2+9100Series/article8075.htm

References (6)

[46]: Stackhouse B. et al., A 65 nm 2-Billion Transistor Quad-Core Itanium Processor, IEEE Journal of Solid-State Circuits, Vol. 44, No. 1, Jan. 2009 http://www.ece.ncsu.edu/asic/ece733/2009/docs/Itanium.pdf

[47]: Schaelicke L., DeLano E., Intel Itanium Quad-Core Architecture for the Enterprise, 2010, http://www.cgo.org/cgo2010/epic8/slides/schaelicke.pdf

[48]: Pawlowski S.: Intelligent and Expandable High- End Intel Server Platform, Codenamed Nehalem-EX, IDF 2009, http://blogs.intel.com/wp-content/mt-content/com/technology/Nehalem-EX_Steve_Pawlowski_IDF.pdf

[49]: Kowaliski C., Intel unleashes quad-core Itanium 9300, Tech Report, Febr. 9 2010, http://techreport.com/news/18445/intel-unleashes-quad-core-itanium-9300

[50]: Power Management in Intel Architecture Servers, White Paper, April 2009, http://download.intel.com/support/motherboards/server/sb/power_management_of_intel_architecture_servers.pdf

[51]: Intel 7500 Chipset, Datasheet, March 2010, http://www.intel.com/content/dam/doc/datasheet/7500-chipset-datasheet.pdf

[52]: The Intel Itanium Processor 9300 Series, White Paper, 2010, http://download.intel.com/pressroom/archive/reference/Tukwila_Whitepaper.pdf

[53]: DeLano E., Tukwila – a Quad-Core Intel Itanium Processor, HotChips 20, Aug. 26 2008, http://www.hotchips.org/wp-content/uploads/hc_archives/hc20/3_Tues/HC20.26.911.pdf

References (7)

[54]: Riedlinger R., Bhatia R. et al., A 32 nm 3.1 billion transistor, 12 wide issue Itanium Processor for Mission-Critical Servers, ISSCC, 2011, http://ewh.ieee.org/r5/denver/sscs/Presentations/2011_10_Riedlinger.pdf

[55]: Itanium in Mission Critical Computing, March 14 2012, http://www.slideshare.net/HP_ESSN_Philippines/mission-critical-computing-by-intel

[56]: Intel Itanium Processor 9500 Series Reference Manual, Software Development and Optimization Guide, July 2012

[57]: Shilov A., Intel Launches Eight-Core Itanium 9500 "Poulson" Mission-Critical Server Processo Xbit Labs, Nov. 8 2012, http://www.xbitlabs.com/news/cpu/display/20121108120233_ Intel_Launches_Eight_Core_Itanium_9500_Poulson_Mission_Critical_Server_Processor.html

[58]: Rubinfeld P., Gianos C., A Detailed Look at the Future Itanium Processor, Openvms Bootcamp, 2011, http://c.ymcdn.com/sites/www.connect-community.org/resource/ resmgr/2011_boot_camp_presos3/p_rubenfeld_intel_future_of_.pdf

[59]: Riedlinger R., Arnold R., Biro L, Bowhill B., A 32 nm, 3.1 Billion Transistor, 12 Wide Issue Itanium Processor for Mission-Critical Servers, IEEE Journal of Solid-State Circuits, Vol. 47, Issue 1, Jan. 2012

[60]: Riedlinger R., Bhatia R. Biro L. et al., A 32 nm, 3.1 billion transistor, 12 wide issue Itanium Processor for Mission-Critical Servers, Technical Paper, July 2011, Rev. 1.1, http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/itanium-poulson-isscc-paper.pdf

References (8)

[61]: McInerney R., Unleashing the Intel Itanium Processor 9500 Series, 2012, http://download.intel.com/newsroom/archive/Intel_Itanium_9500_launch_presentation.pdf

[62]: Niccolai J., Microsoft ending support for Itanium, ComputerWorld, April 4 2010, http://www.computerworld.com/s/article/9174798/Microsoft_ending_support_for_Itanium

[63]: Oracle Stops All Software Development For Intel Itanium Microprocessor, March 22 2011, http://www.oracle.com/us/corporate/press/346696

[64]: Whittaker Z., Oracle to continue Itanium support for HP after court ruling, ZDNet, Sept. 5 2012, http://www.zdnet.com/oracle-to-continue-itanium-support-for-hp-after-court-ruling-7000003743/

[65]: Pollice M., Itanium Gets the Final Nail in the Coffin, BSN, Febr. 20 2013, http://www.brightsideofnews.com/news/2013/2/20/itanium-gets-the-final-nail-in-the-coffin.aspx

[66]: Muljono H., Rusu S., Cherkauer B., Stinson J., New 130nm Itanium 2 Processors for 2003, http://www.hotchips.org/wp-content/uploads/hc_archives/hc15/3_Tue/13.intel.pdf

[67]: DeMone P., Sizing up the Super Heavyweights, Real World Technologies, Oct. 4 2004, http://www.realworldtech.com/super-heavyweights/4/

[68]: Cleveland S., x86-64 Technology White Paper, Advanced Micro Devices, Inc., http://people.cs.clemson.edu/~mark/464/x86-64_wp.pdf

[69]: Hruska J., Itanium's Last Hurrah: Intel Releases the 9700 Series as the CPU Finally Dies, ExremeTech, May 11, 2017, https://www.extremetech.com/computing/249190-itaniums-last-hurrah-intel-releases-itanium-9700-series-cpu-finally-officially-dies

References (9)

[70]: Cutress I., Intel's Itanium Takes One Last Breath: Itanium 9700 Series CPUs Released, AnandTech, May 11, 2017, https://www.anandtech.com/show/11372/intels-itanium-takes-one-last-breath-9700-series-released

[71]: Intel Itanium Processor 9300 Series, 9500 Series and 9700 Series Datasheet, Mai, 2017, https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/9300-9500-9700-series-datasheet.pdf