SGI® Altix™ Hardware Architecture

Reiner Vogelsang
SGI GmbH
reiner@sgi.com

November 5, 2005

SGI Altix 3000

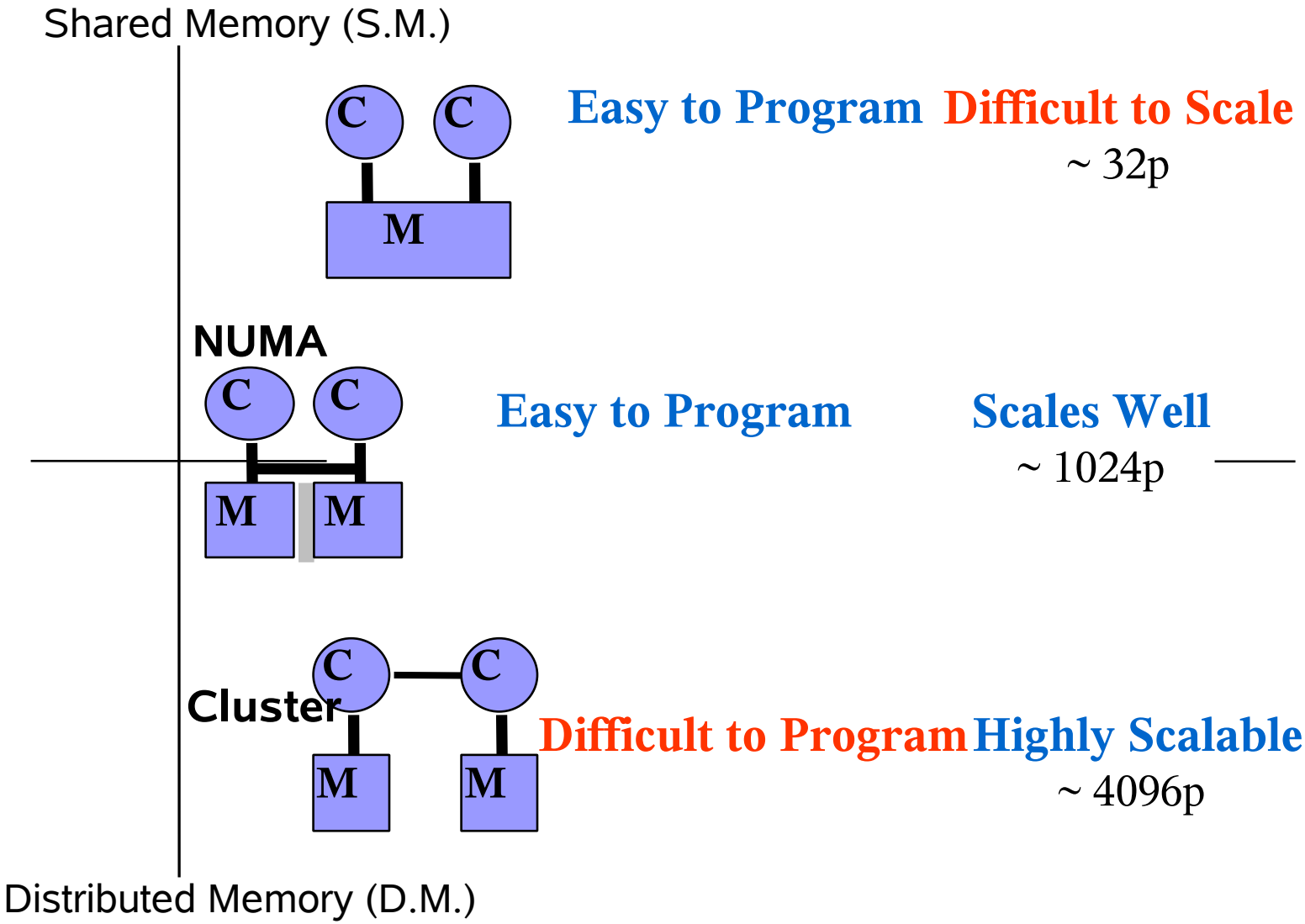
- Introduced January 2003
  - Red Hat- or Suse-Linux compatible Operating System:
  - 512 CPU SSI Linux released
  - Intel Itanium2 processors (Madison) in all variants
  - Over 55000 processors sold, systems from 2 PEs to >512 PEs
  - Huge shared memory
    - Several orders for < 100 PEs with >2 TB shared memory.
  - Two OS Variants:
    - SGI enhanced Red Hat AS 2.1 based Linux OS
    - Standard SUSE SLES 9
  - Most traditional SGI value-adds available:
    - CXFS Client/Server, DMF, MPT,SCSL
    - Multipipe GFX available

SGI ccNuma Balanced System Architecture

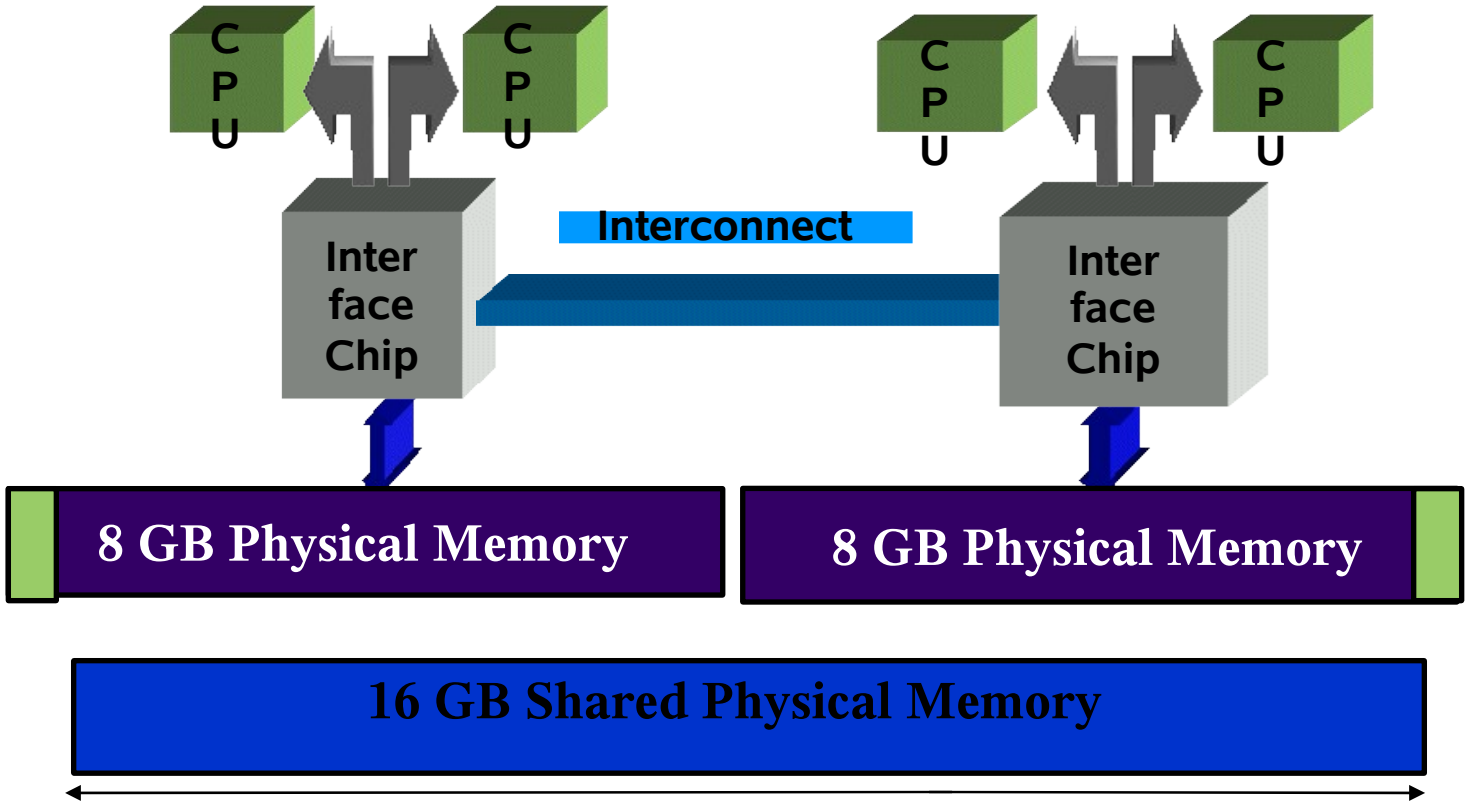
SGI Altix3000

- SGI Altix3000 computer system is characterized by:
  - (Scalable) cache coherent shared memory (SMP)
  - Intel Itanium-2 processors
  - Standard Linux operating system

Parallel Architectures



SGI Scalable ccNUMA Architecture

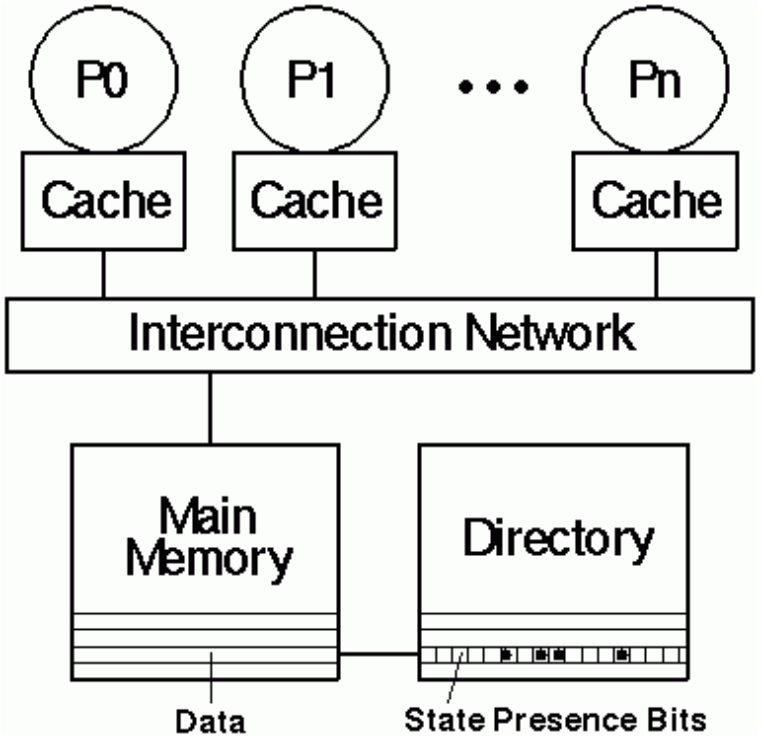


ccNuma: Distributed Shared Memory

- ccNuma:
  - Memory is physically distributed but logically shared
  - Memory is kept coherent automatically by hardware
  - Coherent memory: memory is always valid (caches hold copies)
  - Granularity is L3 cacheline (128 B)
- Directory memory:
  - For each cacheline access information is stored:
  - Who has valid copies
  - Which processor has write access
  - Hardware revokes access rights automatically
- *In contrast snoopy bus protocols do not scale well*
  - *Access requests are broadcasted*
- Directory information is stored in main memory
  - Directory entry is 4 byte wide for each 128 byte cache line

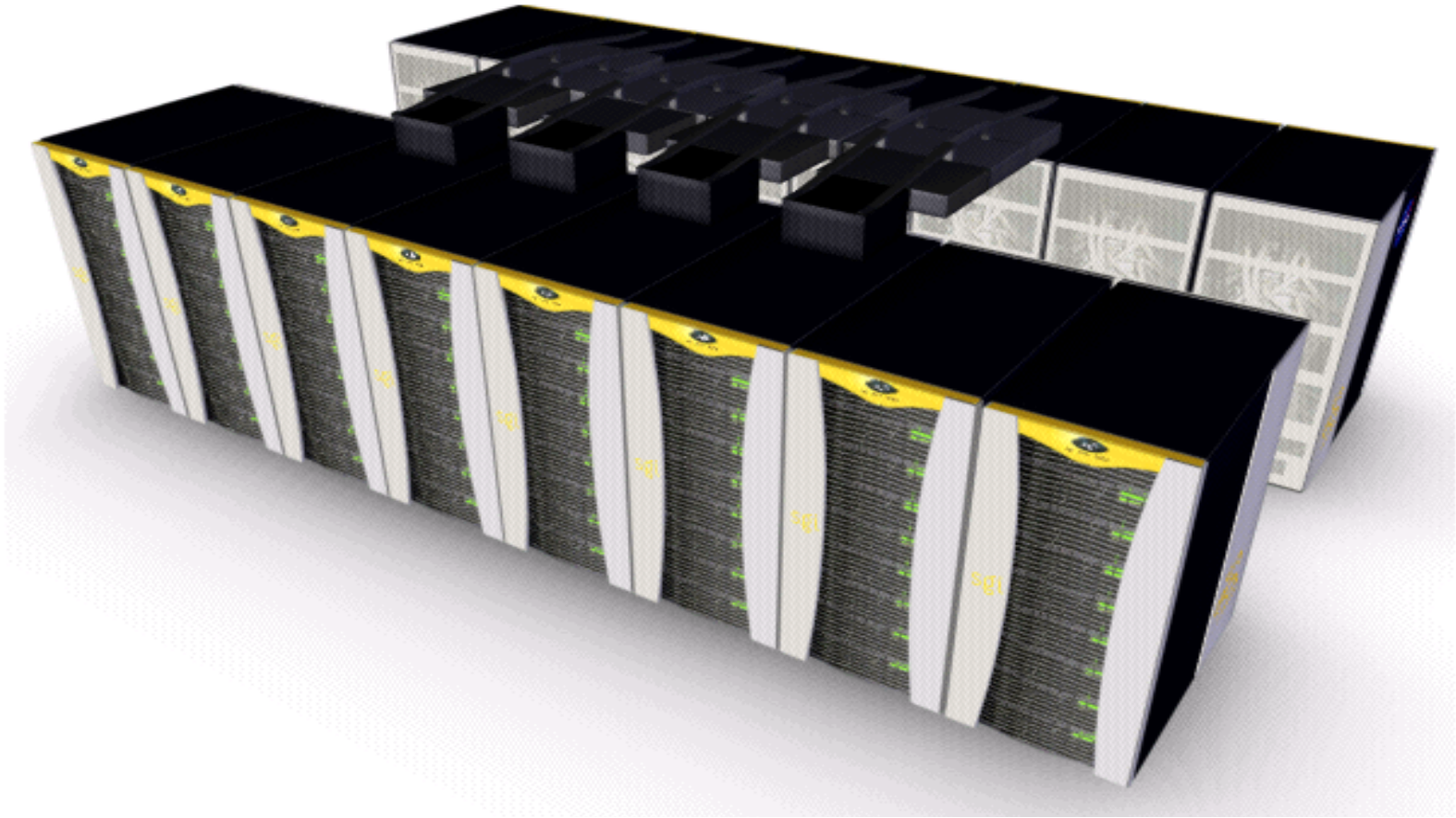
ccNuma: Distributed Shared Memory

- Schematic view onto a full directory based coherence scheme



- The Stanford Dash Multiprocessor," by D. Lenoski et al., IEEE Computer, 25(3), March 1992, pp 63-79”
- http://www.cse.ucsd.edu/classes/fa00/cse240/lectures/Lecture18.html

SGI Altix 3700BX2



SGI® Altix™ Product Family

SGI® Altix™ 3700 Bx2

- 64 processors per rack
- NUMALink 4 (6.4 GB/sec)
- Madison 9M

Altix 1350

SGI® Altix™ 3700

- 32 processors per rack
- NUMALink 3 (3.2GB/sec)
- Madison

SGI® Altix™ 350

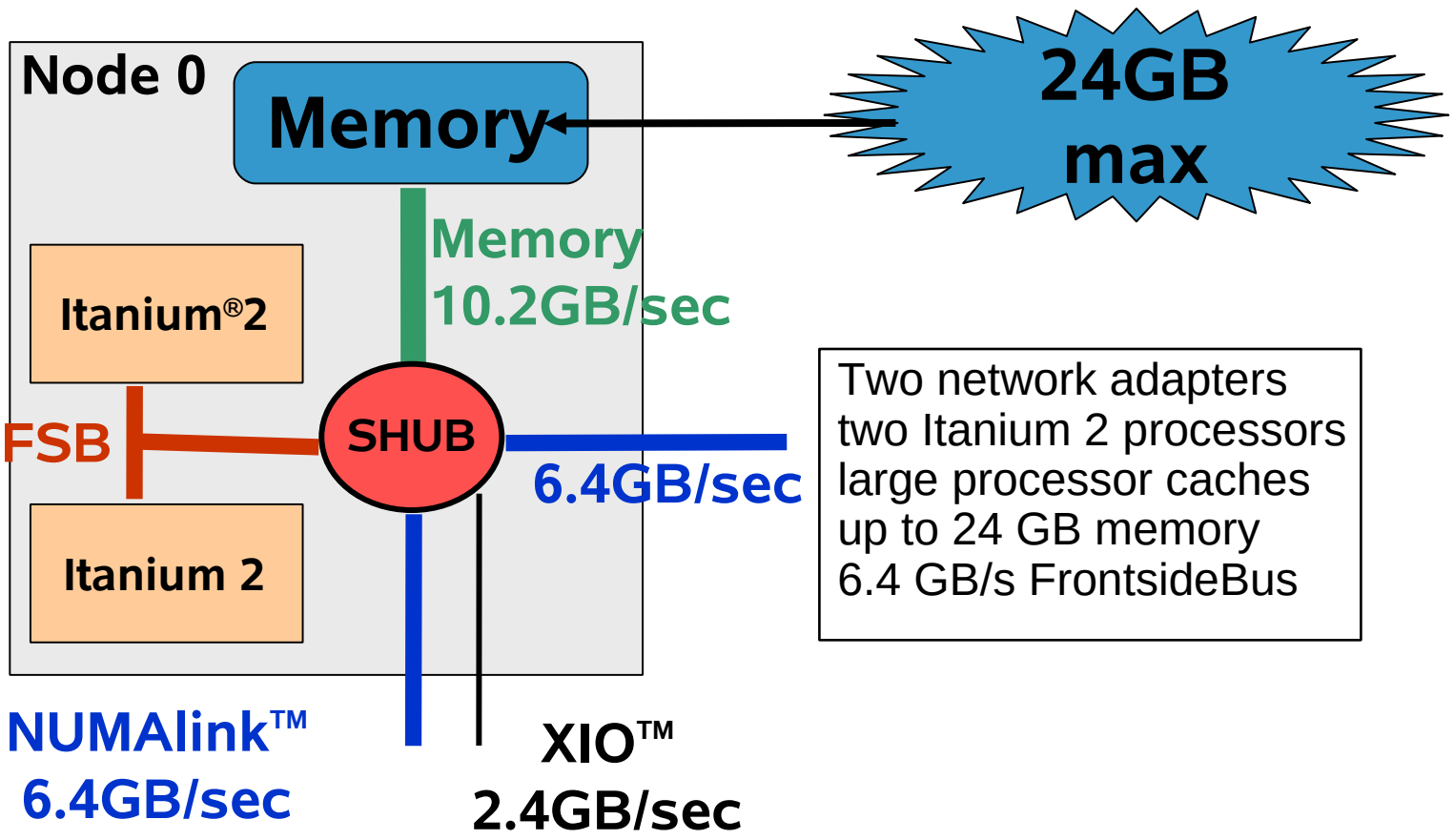
- Mid-range server
- Scales up to 32P
- Modular "expand on demand" architecture

SGI® Altix™ 4700

- 128++ processors per rack
- NUMALink ++
- Blades
- Montecito

System Scalability

SGI® Altix™ 3000BX2 CPU-Module



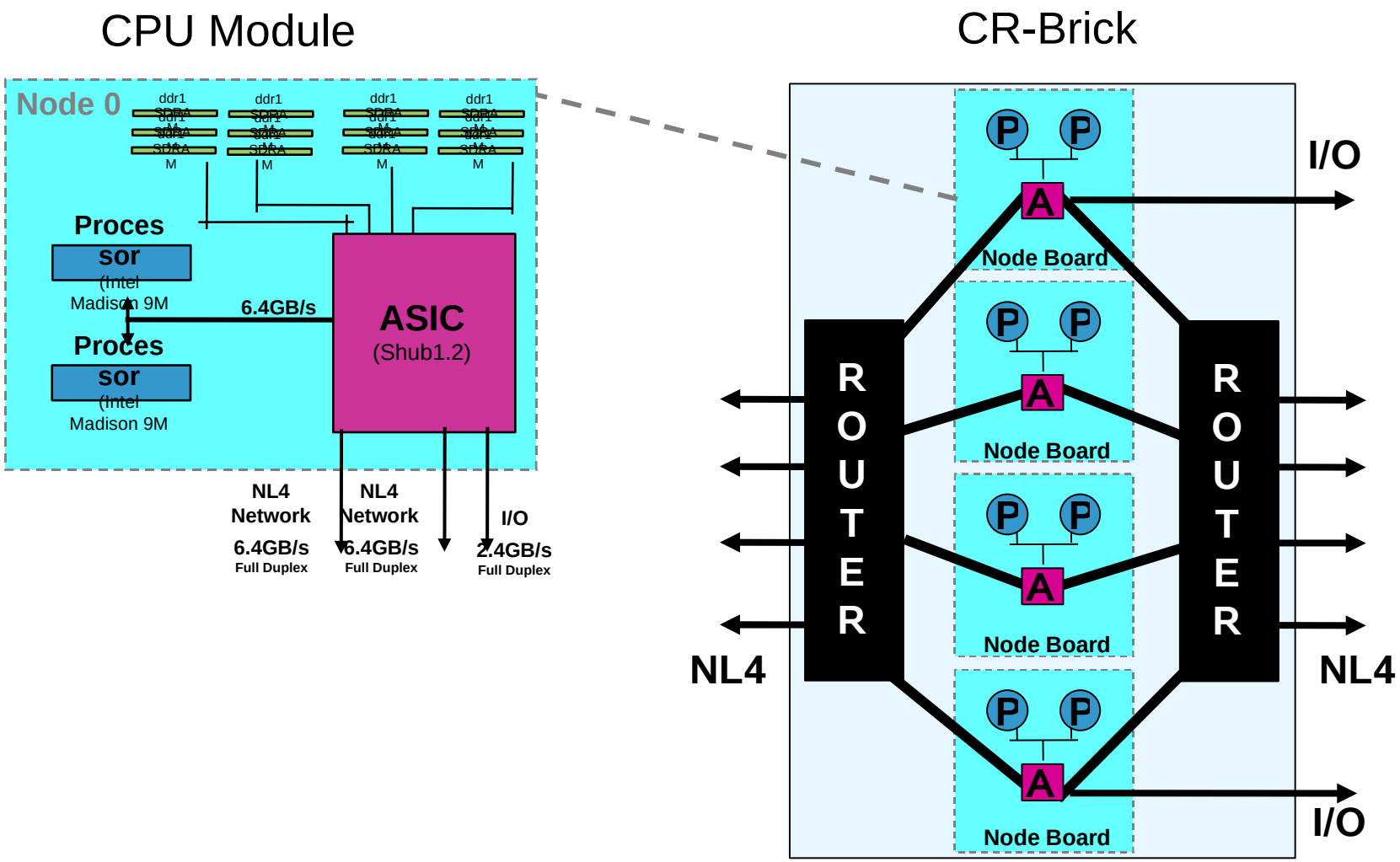
Two network adapters
two Itanium 2 processors
large processor caches
up to 24 GB memory
6.4 GB/s FrontsideBus

SGI® Altix™ 3000BX2 Memory

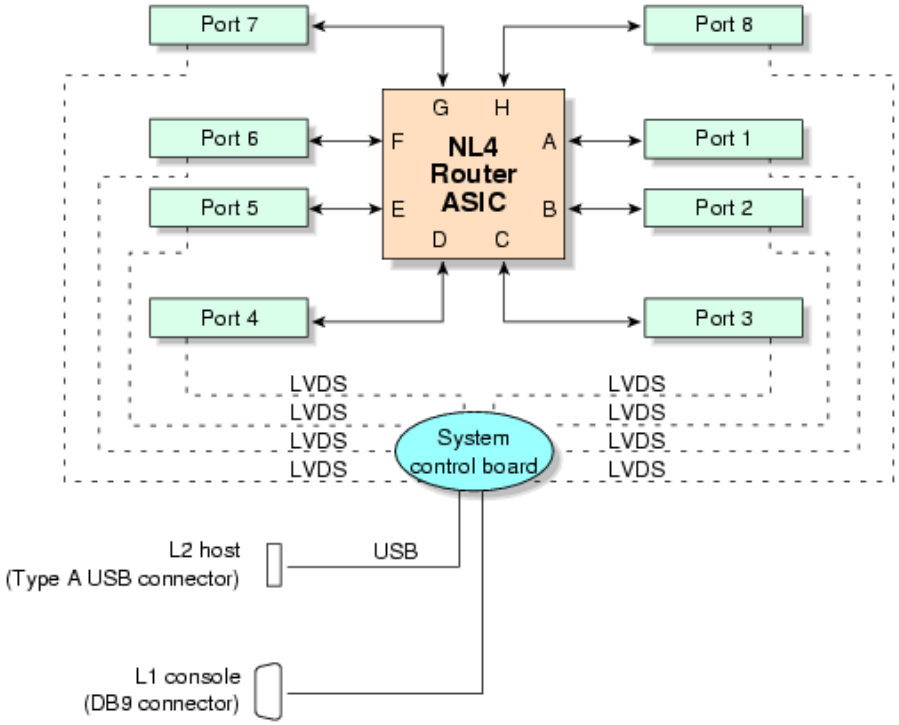
Each CPU module:

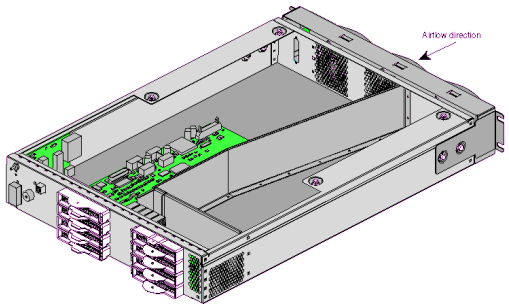
- 4 banks of up to 3 DDR-SDRAM dimms
- Dimms are 512 MB, 1GB or 2GB in size
  - PC2100 = 133MHz (DDR226) Altix BW = 8.5 GB/s - 7.5 ns
  - PC2700 = 166MHz (DDR333) Altix BW = 10.2 GB/s - 6.0 ns
  - PC3200 = 200MHz (DDR400) Altix BW = 12.8 GB/s - 5.0 ns

SGI Altix™ 3700 Bx2 Platform Introduction:
CR-Brick - Components



Router

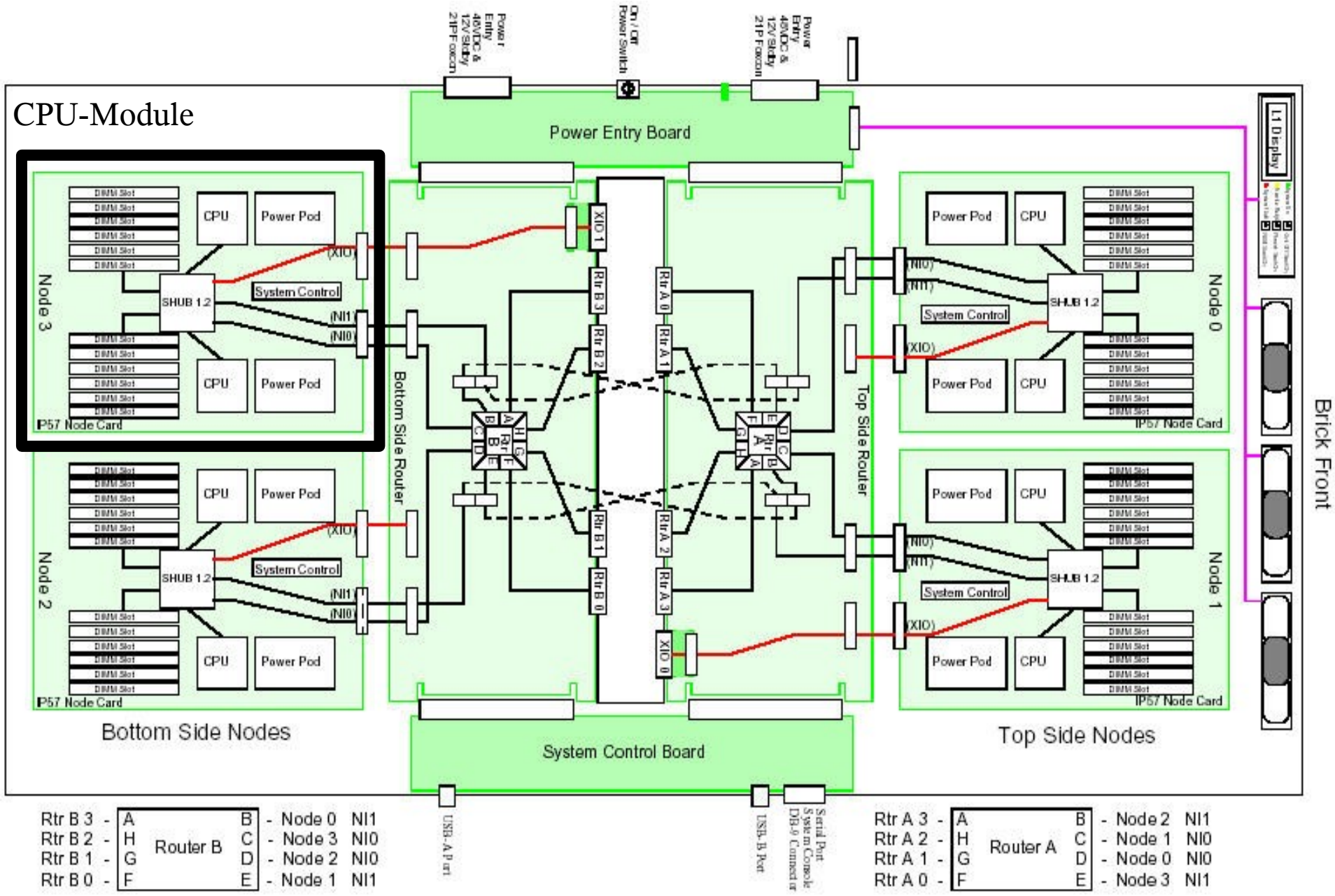




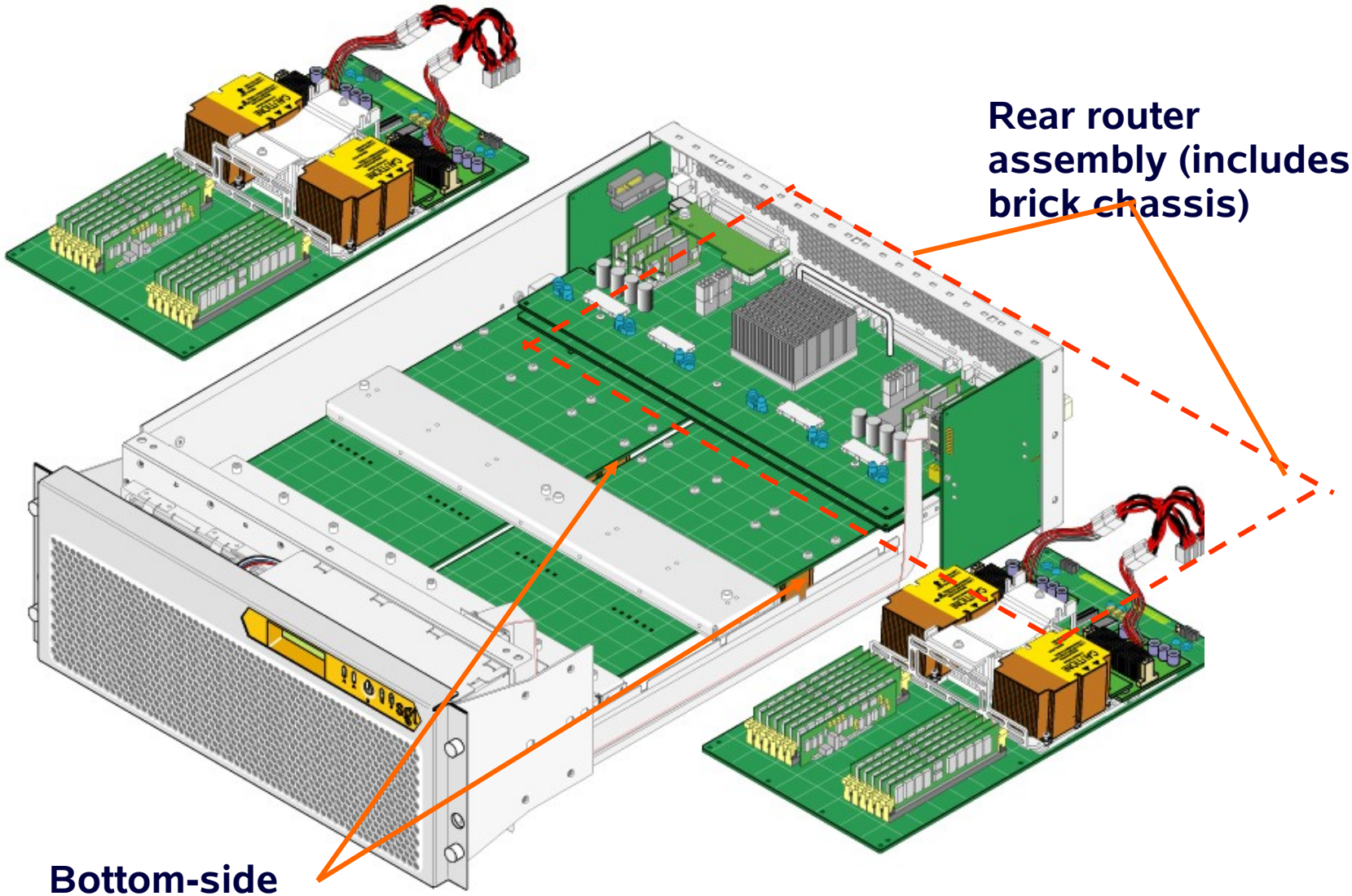
Numaflex-4 Router:

- **Microarchitecture elements of Cray T3E**
  - **Enhanced hardware support synchronization primitives**
- **8 bidirectional ports**
- **3.2 GB/s per direction per port**
- **Low latency about 50 nsec per router**
- **Dual plane configuration:**
  - **2 x 6.4GB/sec total bandwidth between C-bricks**

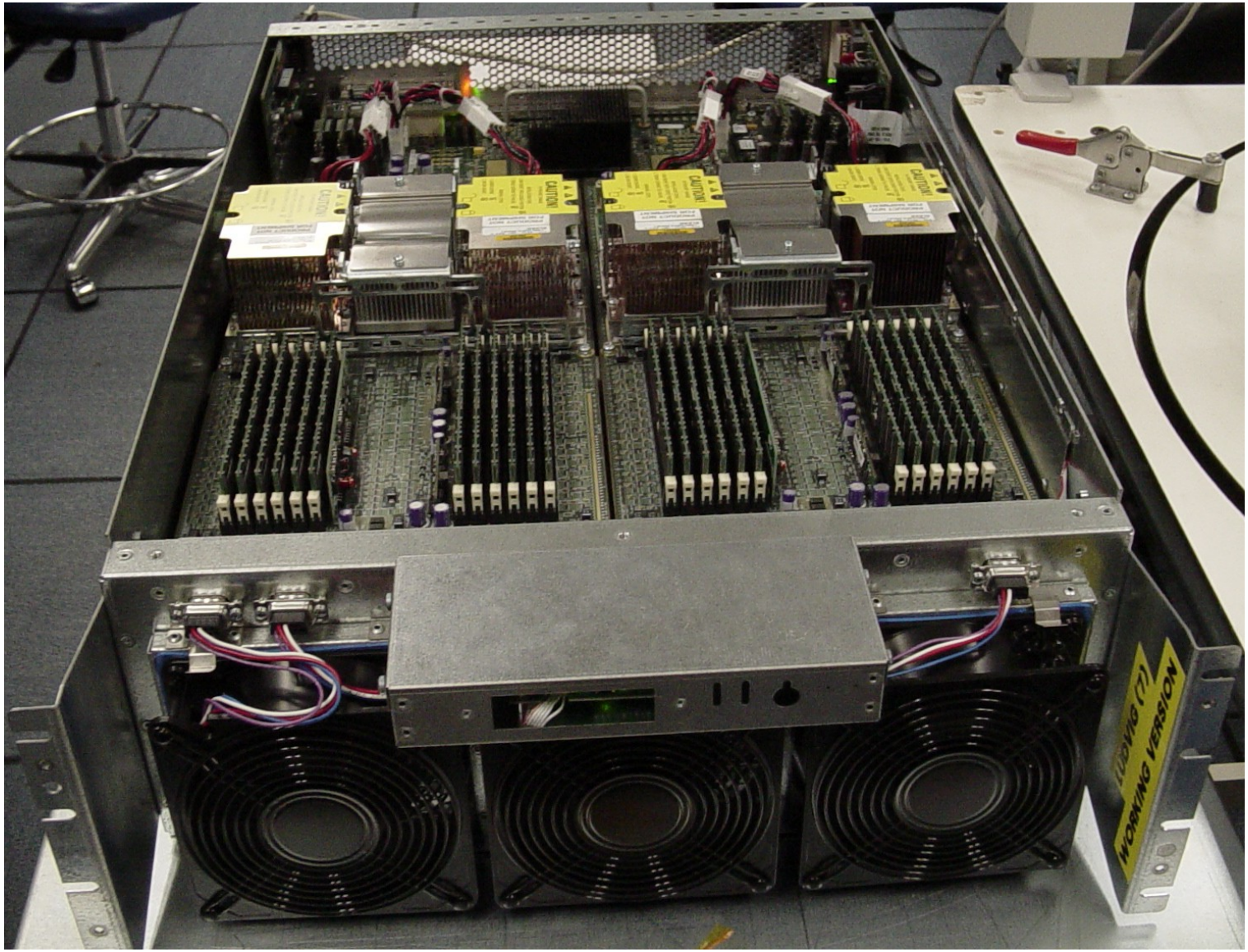
SGI Altix 3700 BX2 CR-Brick



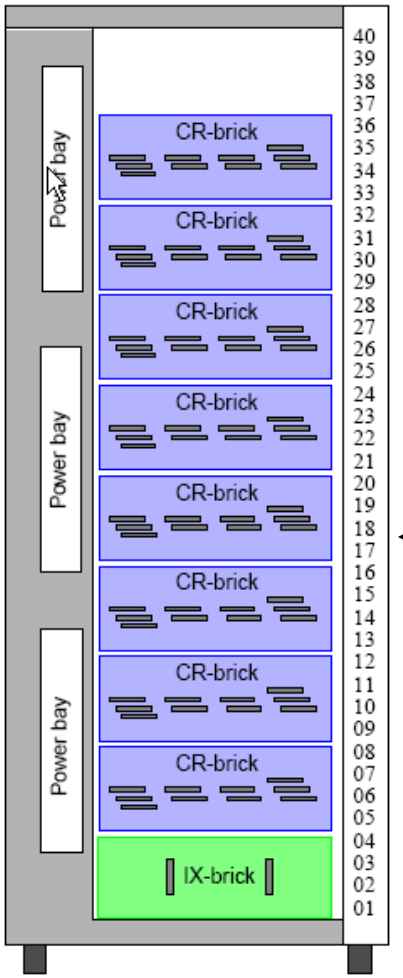
SGI Altix 3700 BX2 CR-Brick



SGI Altix 3700 BX2 CR-Brick

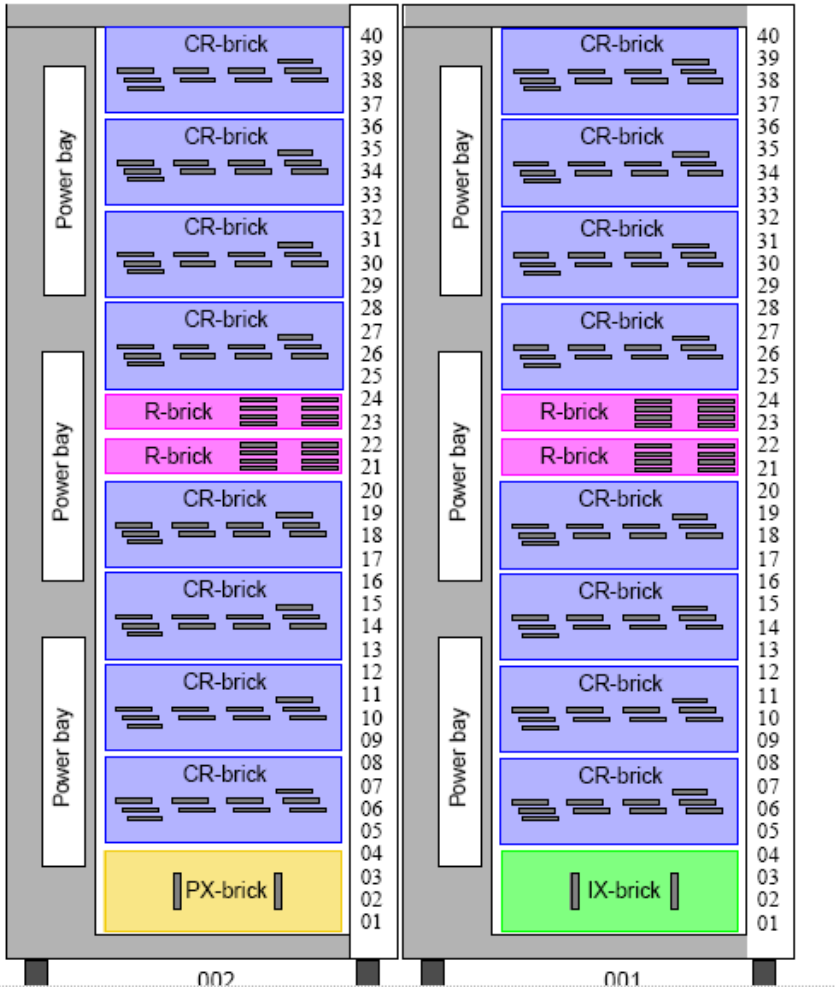


Altix3700BX2



Note:
Additional D-brick racks not shown.

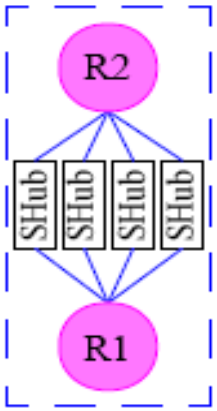
Altix 3700



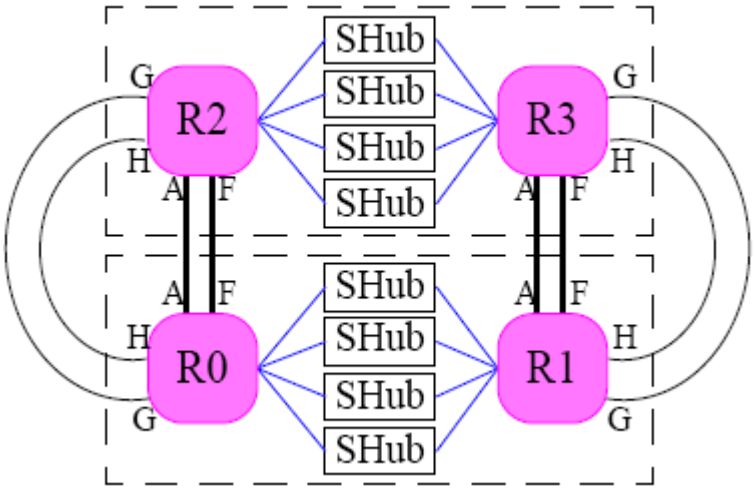
Note:
Additional racks with D-bricks not shown.

SGI Altix BX2 Configurations

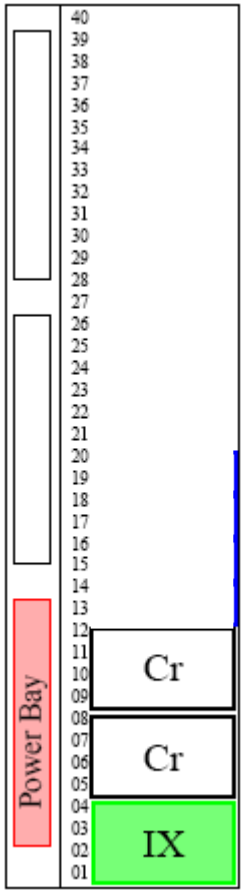
Configurations



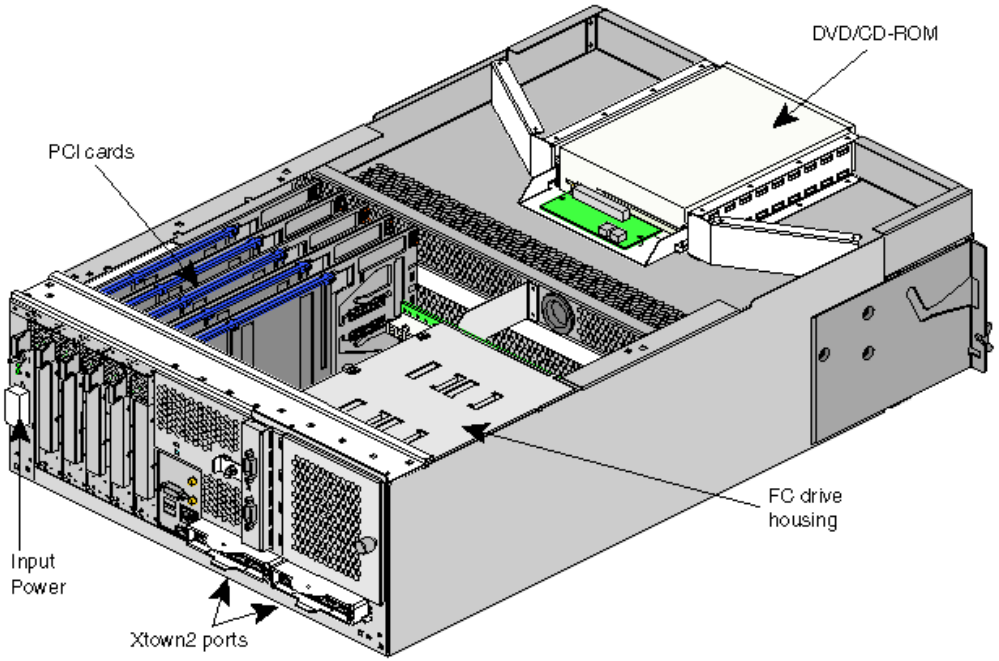
8 CPU System



16 CPU System



IX-Brick



- **IX-brick**

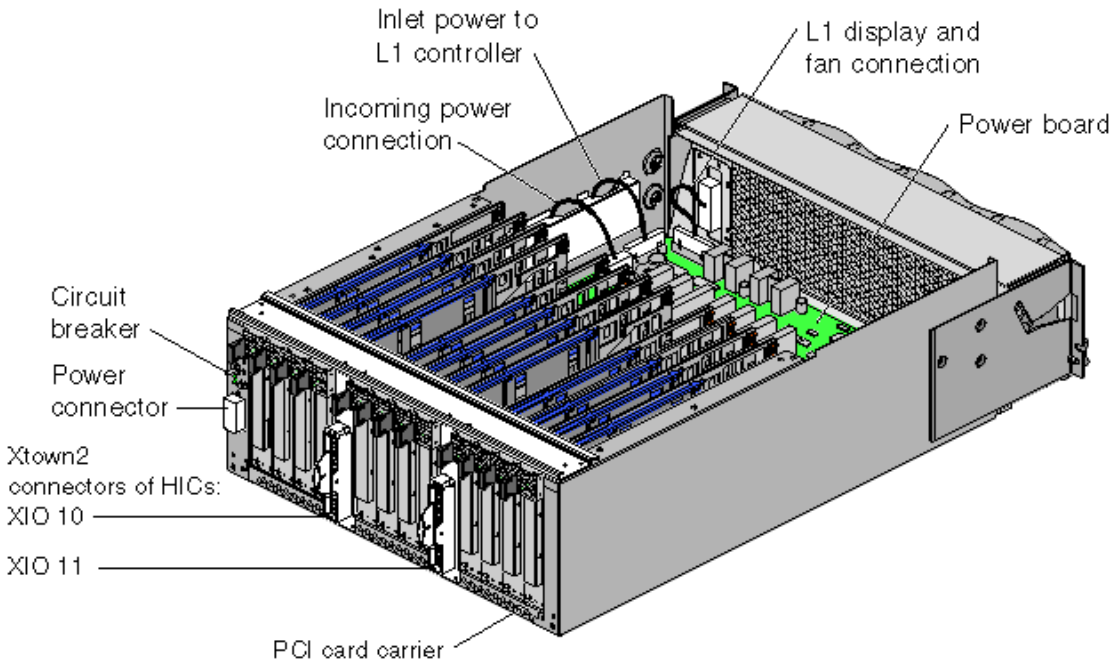
Function

- provides basic I/O functionality, nouses the system disk and one option drive maximum
- DVD-ROM, 2 serial ports, and 10/100/1000Bt
- 10 PCIX expansion slots on 5 buses + 1 PCI 66MHz slot
- optional 2 serial port expansion card
- 1 per partition required

Connectivity to a C-brick

- 1 or 2 XIO™ + input ports with up to 4.8GB/sec aggregate B/W

PX-Brick



- **PX-brick**

  Function

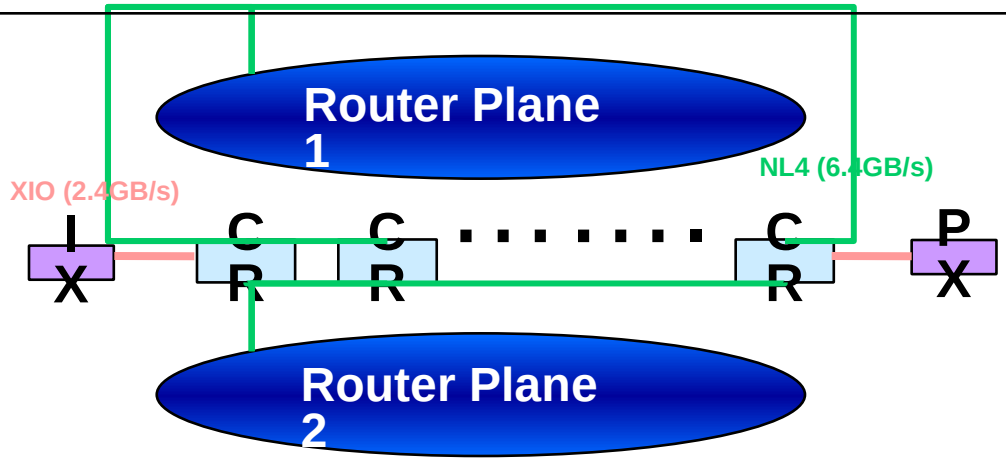
  - **Provides PCIX expansion without basic I/O overhead**
  - **12 PCIX expansion slots on 6 buses**

  Connectivity to a C-brick

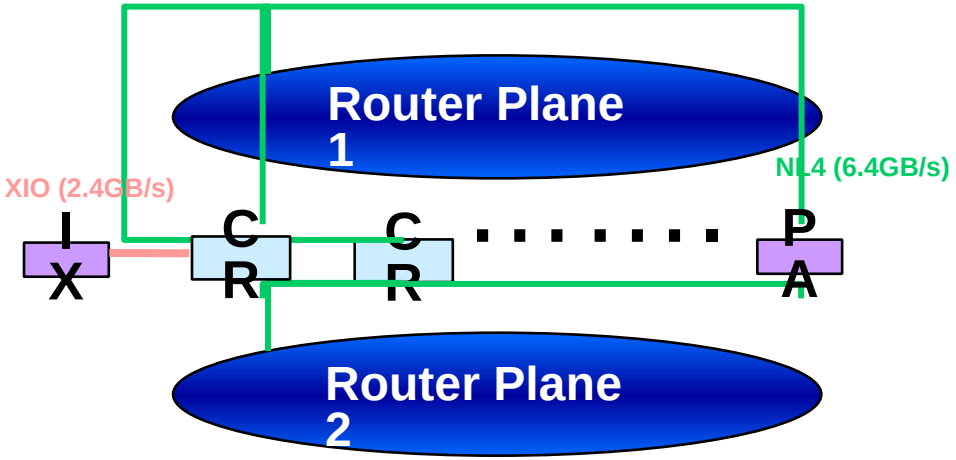
  - **1 or 2 XIO + input ports with up to 4.8GB/sec aggregate B/W**

SGI Altix™ 370 0 BX2 Platform Introduction: I/O
System Topology of CR-Brick to I/O Bricks

SGI Altix 3700 and Bx2 Topology of CR-Bricks with IX, PX-Bricks

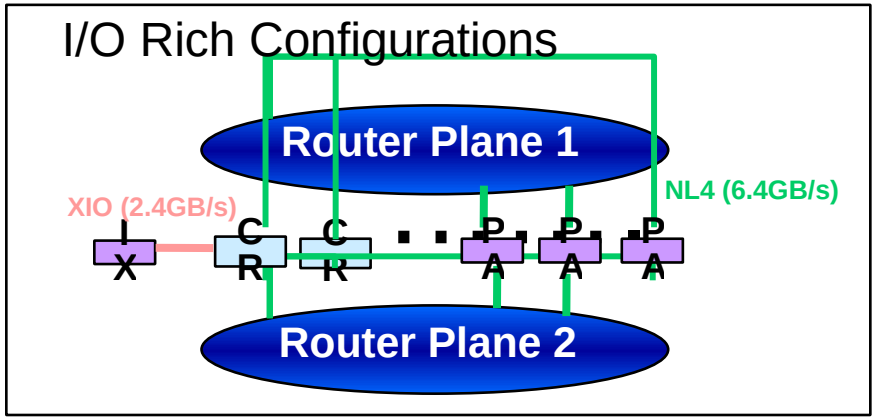


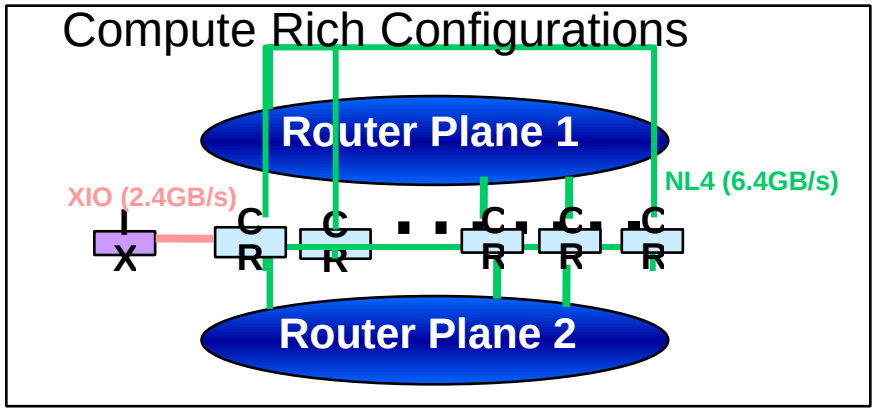
SGI Altix 3700 Bx2 Topology of CR-Bricks with PA-Brick (Peer I/O)



SGI Altix™ 3700 Bx2 Platform Introduction:
Benefits of Peer I/O

- Available with the PA-Brick on SGI Altix 3700 Bx2 Platform
- Direct connection of I/O into NUMAlink fabric:
  - NL4 at 6.4GB/s vs XIO at 2.4GB/s Total flexibility in ratio of compute to I/O capability
- Allows I/O channel performance to scale concurrently with NUMAlink improvements





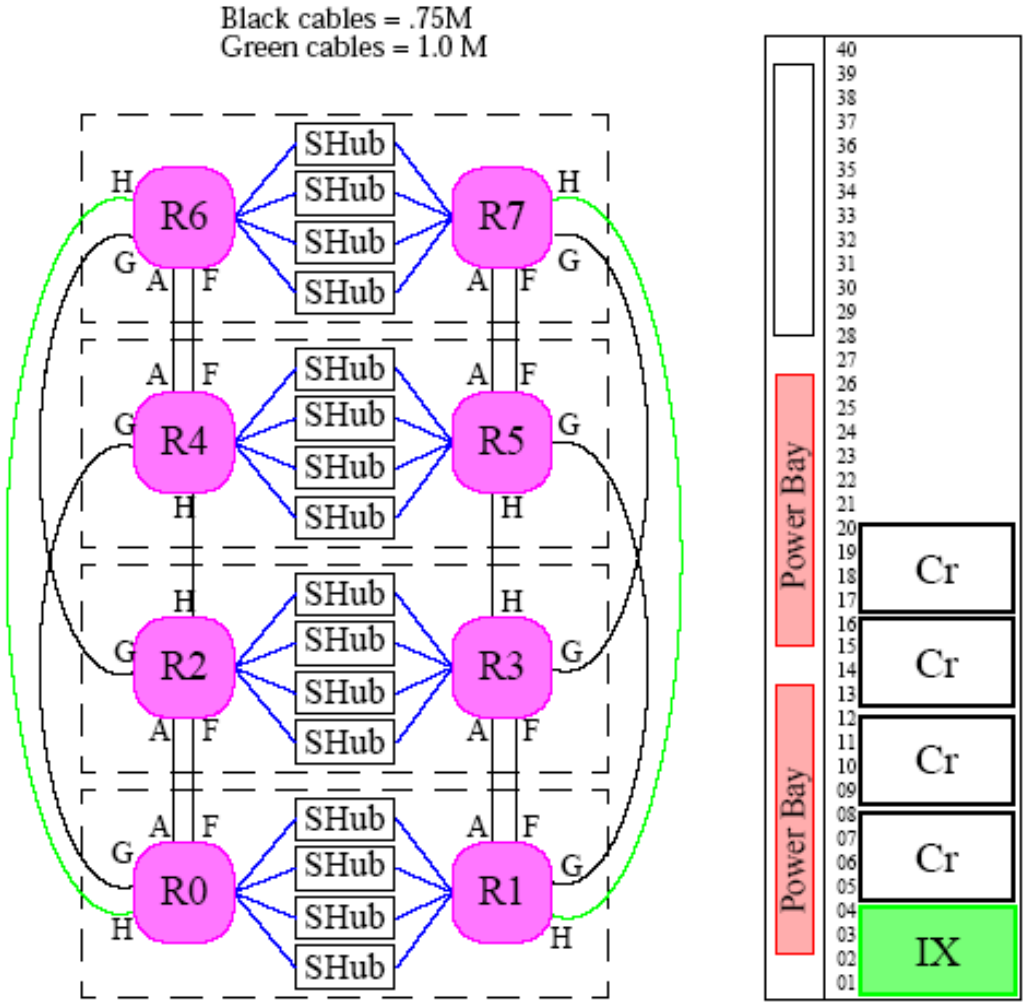
Excursion on PCI

- Peripheral Component Interconnect
  - Invented by Intel
  - Started as 32-bit bus
  - Bus is buffered and works asynchronously
  - Supports Plug and Play configuration (PnP)
- PCIX, extension to width of 64 bits, up to 133 Mhz
- Some performance data

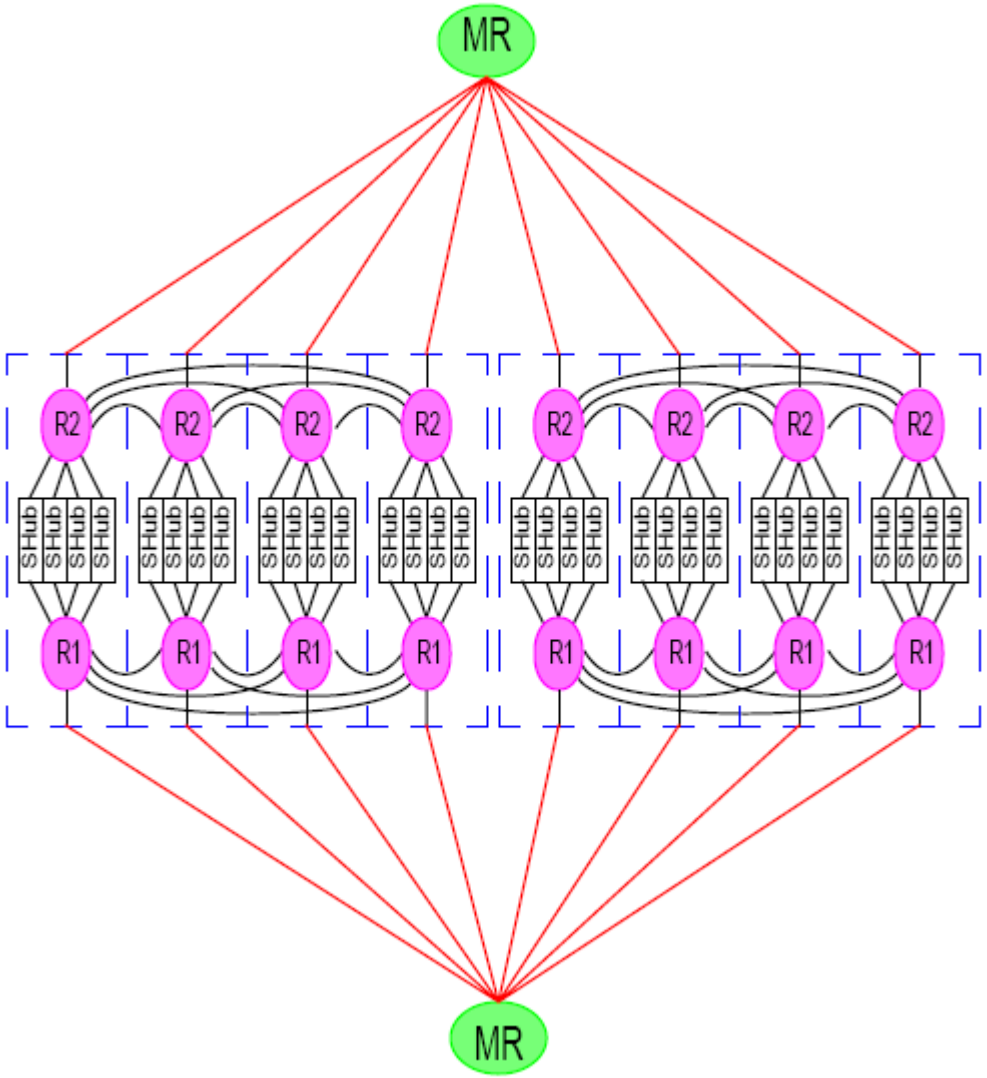
PCI		PCI-X		
33 MHz	66 MHz	66 MHz	100 MHz	133 MHz
132 MB/s	256 MB/s	512 MB/s	800 MB/s	1000 MB/s

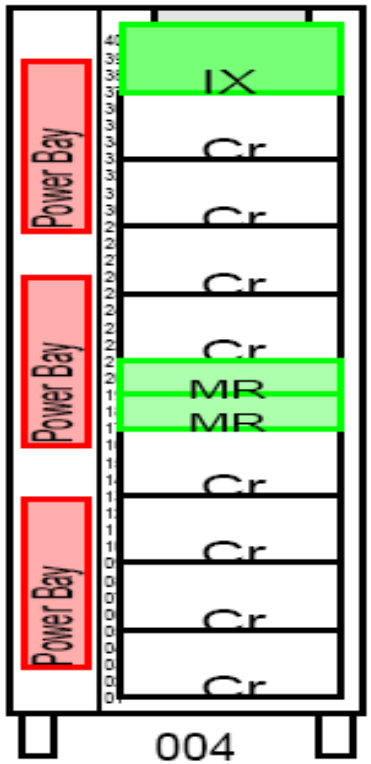
- http://www.pcisig.com/specifications/
- http://arstechnica.com/articles/paedia/hardware/pcie.ars/1

32-CPU Konfiguration

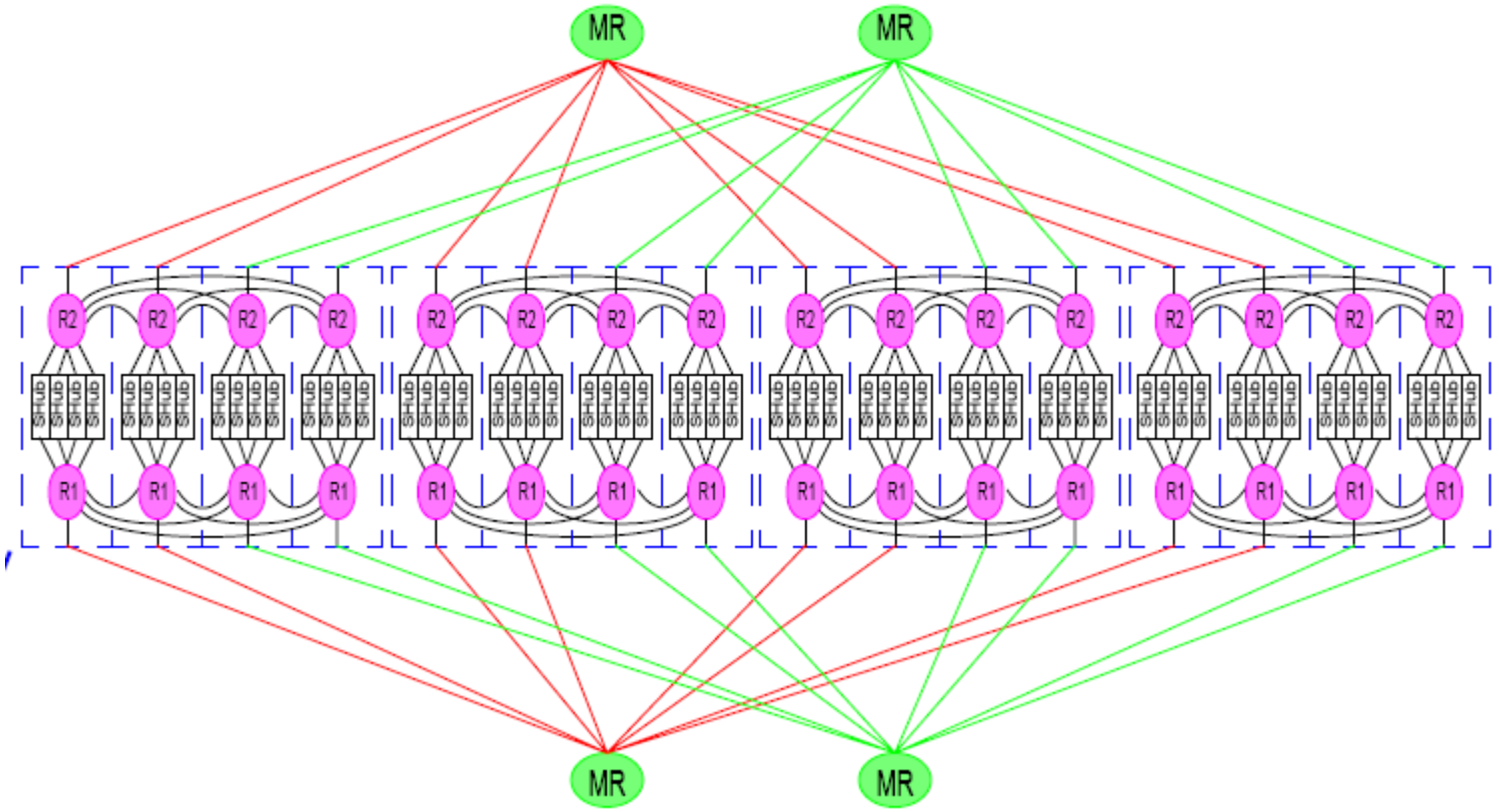


64-CPU Configuration

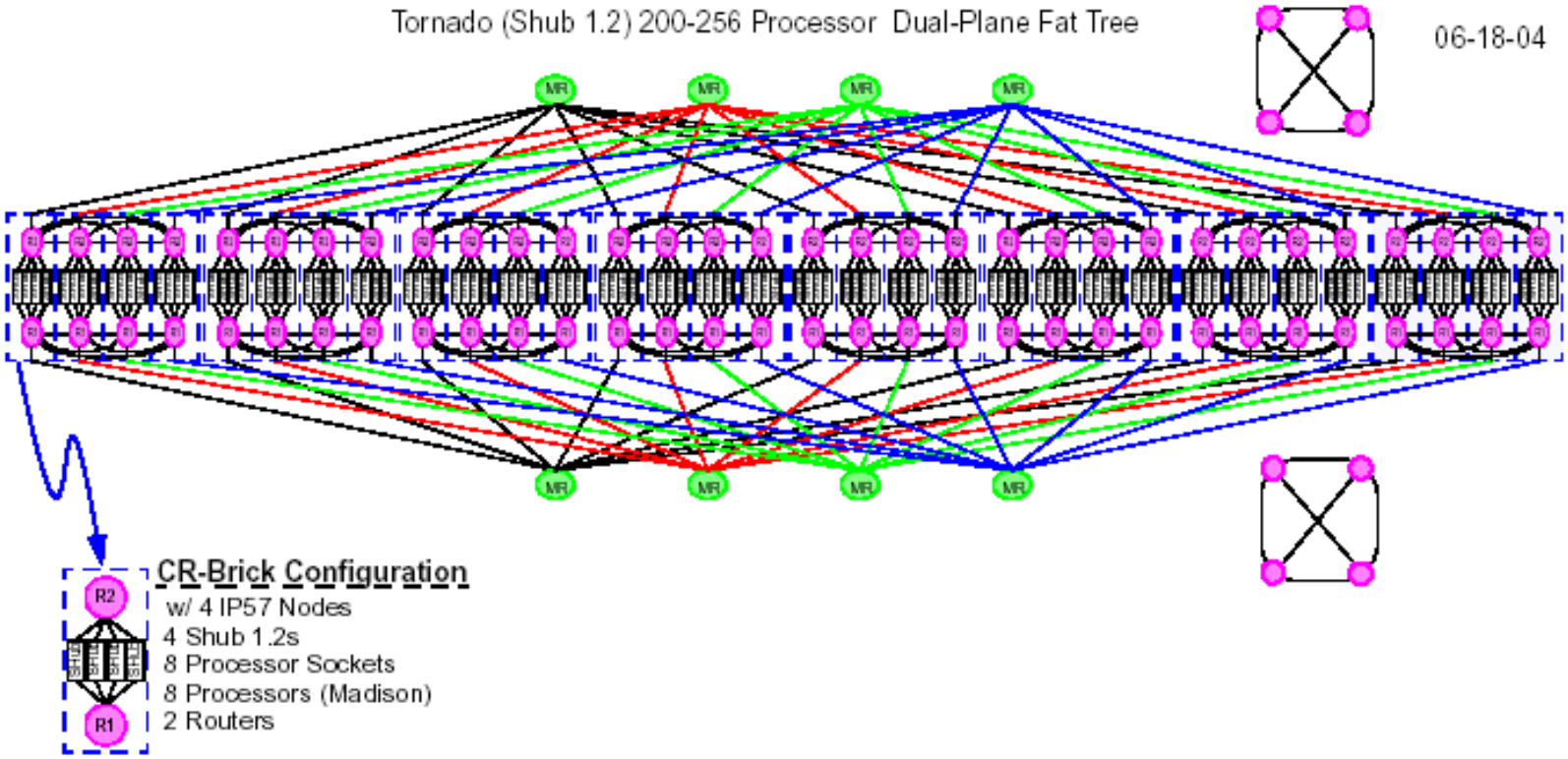




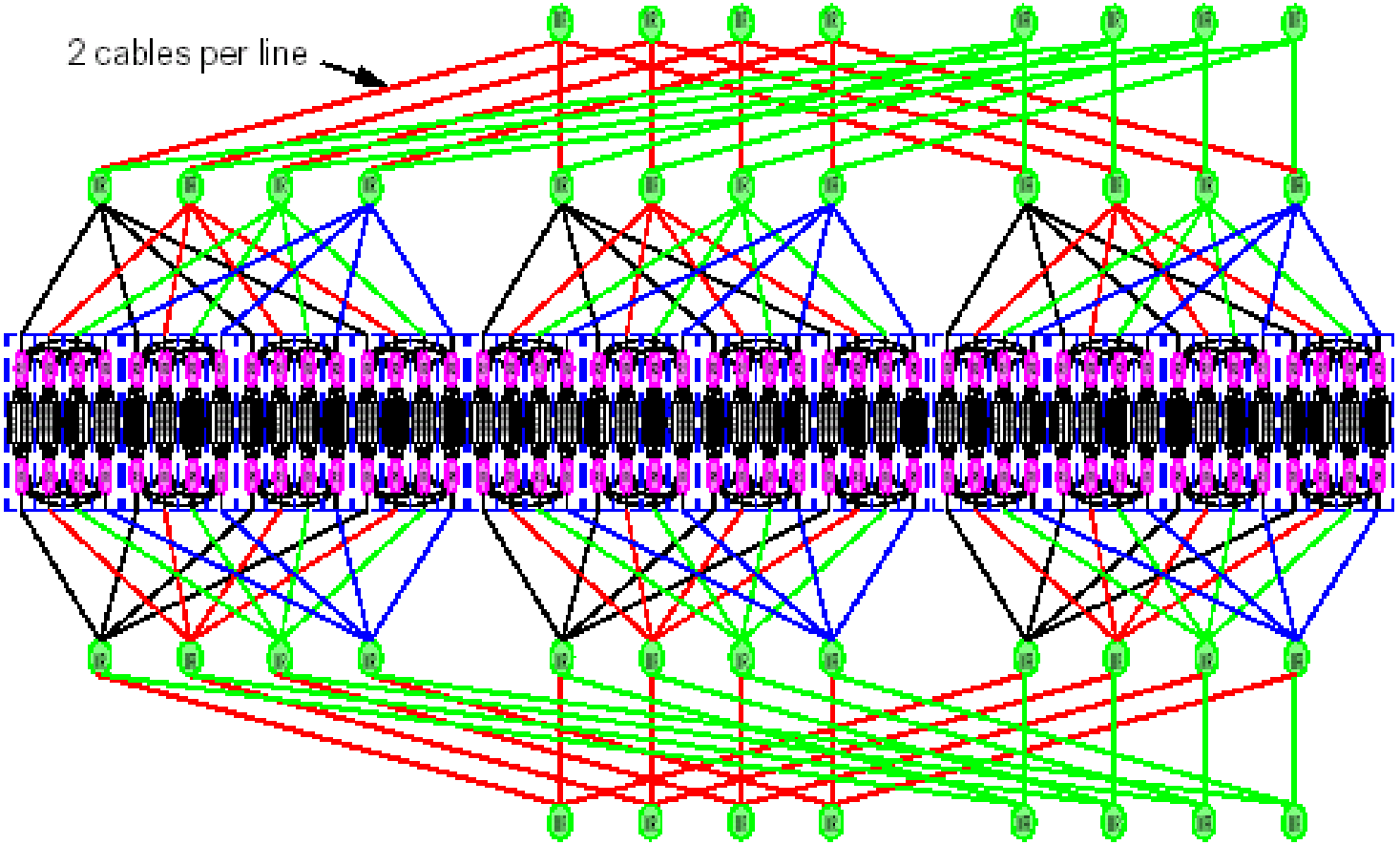
128-CPU Configuration



256 CPUs

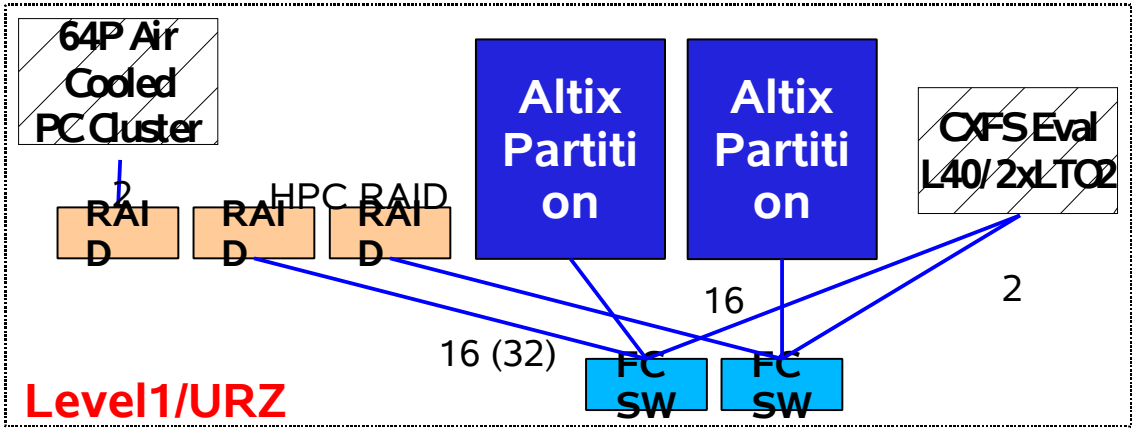

Tornado (Shub 1.2) 200-256 Processor Dual-Plane Fat Tree
06-18-04
MR
CR-Brick Configuration
w/ 4 IP57 Nodes
4 Shub 1.2s
8 Processor Sockets
8 Processors (Madison)
2 Routers
R2
R1

384 CPUs



Altix 3700 BX2 at TU Dresden

Altix 3700 BX2 at TU Dresden



- Two partitions
  - venus, 128 CPUs
  - merkur, 64 CPUS


intel inside
ITANIUM 2

Intel® Itanium® 2 - Why it is important?

High Bandwidth

Many functional units

Large onchip caches

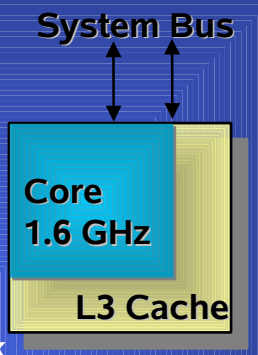
Large physical address space

System Bus
128 bits wide
200 MHz/400 MT/sec
6.4GB/sec

Width
2 bundles per clock
6 integer units
2 loads and 2 stores per clock
11 issue ports
4 FPMultiply Adds per Clock

Caches
L1: 2X16KB—1 clock latency
L2: 256K—5 clock latency
L3: 3-9MB—12 clk
32GB/sec bandwidth

Addressing
50-bit physical addressing
64-bit virtual addressing
Maximum page size of 4GB



Explicitly Parallel Instruction Computing (EPIC)

- EPIC
  - New instruction set (with IA-32™compatibility)
  - 3 predicated instructions into 1 bundle (128bit)
  - 2 bundles per cycle
  - 128 general (integer) registers; up to 96 rotating
  - 128 floating-point registers; up to 96 rotating
  - 64 1-bit predicate registers; up to 48 rotating
  - 8 branch registers
  - 128 application registers (e.g., loop or epilog counters for pipelining)
  - Performance Monitor Unit (PMU) (> 100 Performance Counters)
  - Advanced Load Address Table (ALAT)
  - 6 integer units
  - 2 loads and 2 stores per clock cycle, speculative loads
  - 11 issue ports
  - Special instructions (multimedia, popcnt)

IA-64™ Instruction Bundles

1 instruction coded on 41 bits
3 instructions grouped into 1 bundle (128 bits)

Bundle type is specified through 5-bit template :

```
{       .mfi                                    // template (mem-fp-int)
   (p16) ldfd f39=[r2],16                       // load fp, post-increment
   (p19) fnma.d.s0 f49=f42,f6,f45               // multiply Add
   (p16) adds r32=16,r33        };              // integer add immediate
{          .mib                                 // template (mem-fp-br)
   (p16) ldfd f42=[r33]                         // load fp, post-increment
   (p16) adds r40=8,r33
   br.ctop.dptk.few .BB13_mp_ortho2_ ;;     };// counted loop branch
```

IA-64™ Branch Optimization

Predication allows to remove (small) branches:

```
if ( i == j) {
    k += l;
   x = y + a * b;
} else {
    k = m - 3;
    y = * p_fp ++ ;
}
```

```
cmp.eq p1,p2=r32,r33 ;;
(p1) add r1 = r1, r3
(p1) fpma.d f31 = f3, f4, f2
(p2) sub r1 = 3, r4
(p2) ldfd f31=[r34], 8
```

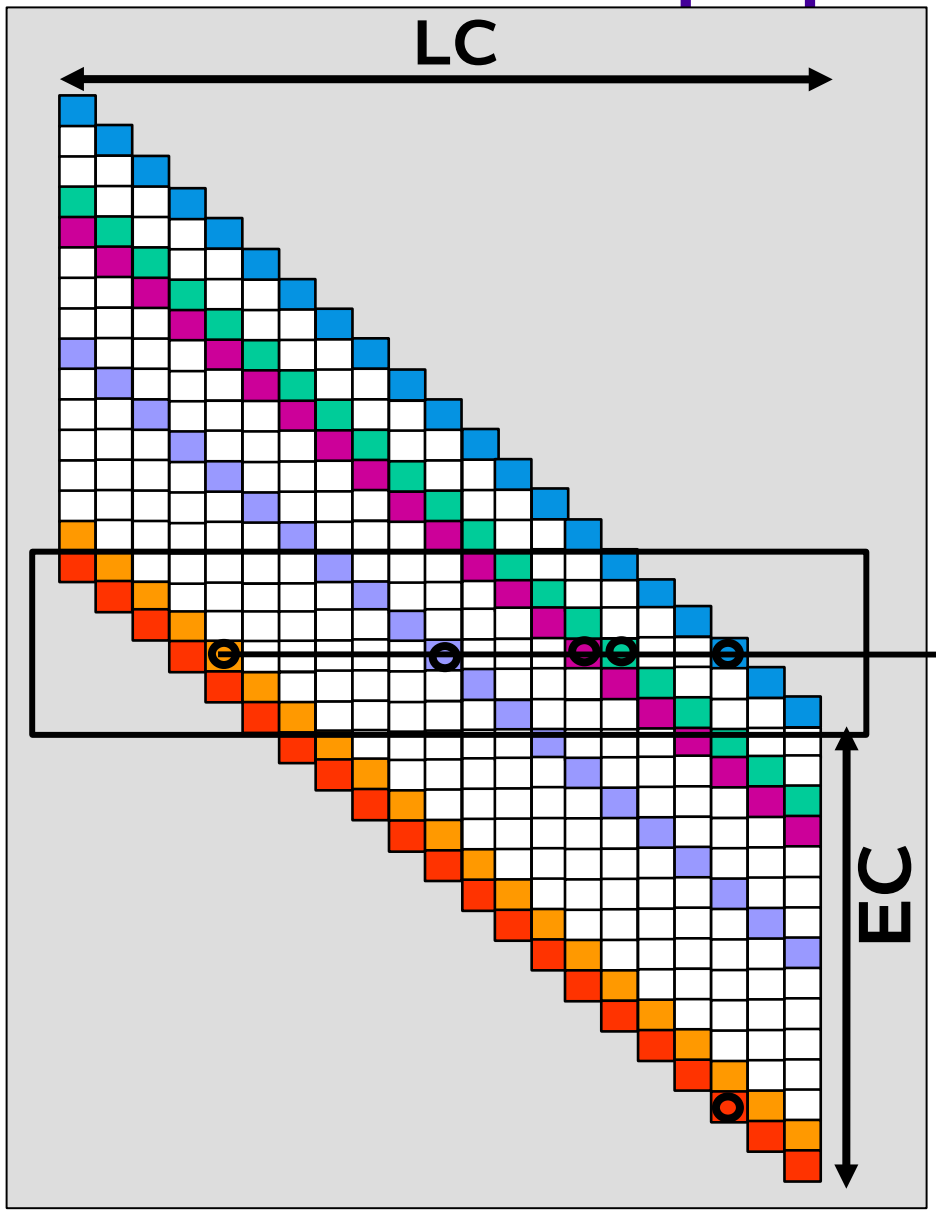
cycle 0
cycle 1
cycle 1
cycle 1
cycle 1

IA-64™ HW for Loop Optimization

Counted loops are optimized with HW support:

- **Loop counter**
- **Epilog counter**
- **Predication registers for each instruction**
- **Rotation of registers**

IA-64™ HW for Loop Optimization




```
{ .mfi
  (P16)
  (P24)
  (P19)      } ;

{ .mib
  (P30)
  (P20)
             br.ctop ;; } ;
```

Itanium™2 - Execution Units

- 6 ALU ALU0-5
- 2 Integer I0,I1
- 1 ISHIFT
- 4 Port Data Cache Unit (2ld[fp]+2st or 4ldf)
- 6 Multimedia PALU0-5
- 2 Parallel shift PSMU0,1
- 1 Parallel Multiply PMUL
- 1 POPCNT
- 2 FP multiply-add FMAC
- 2 FP other operations FMISC
- 3 Branch

Itanium™2 - Instructions Latency

– Integer Instructions	1 cycle
– Floating Point Instructions	4 cycles
– MultiMedia	2 cycles
– FP Multiply-Add/sub *fma/fnma/fms*	4 cycles
– FP Multiply or Add (*fma* x*y+0 or x*1+y)	4 cycles
– no FP Div, use approx[256] *frcpa*	4 cycles
– no FP RSQRT, use approx[256] *frsqra*	4 cycles
– no integer mult, use *setf/xma/getf*	6/4/5 cycles
– no integer Mod, Div use *setf/frcpa/…/getf*	6/4/5 cycles

Itanium™2 - FP Macros Latency

x/y, 1/sqrt(x), sqrt(x) do not translate into HW instructions.

Instead the compiler combines fma/frcpa/frsqra (Newton iterations).

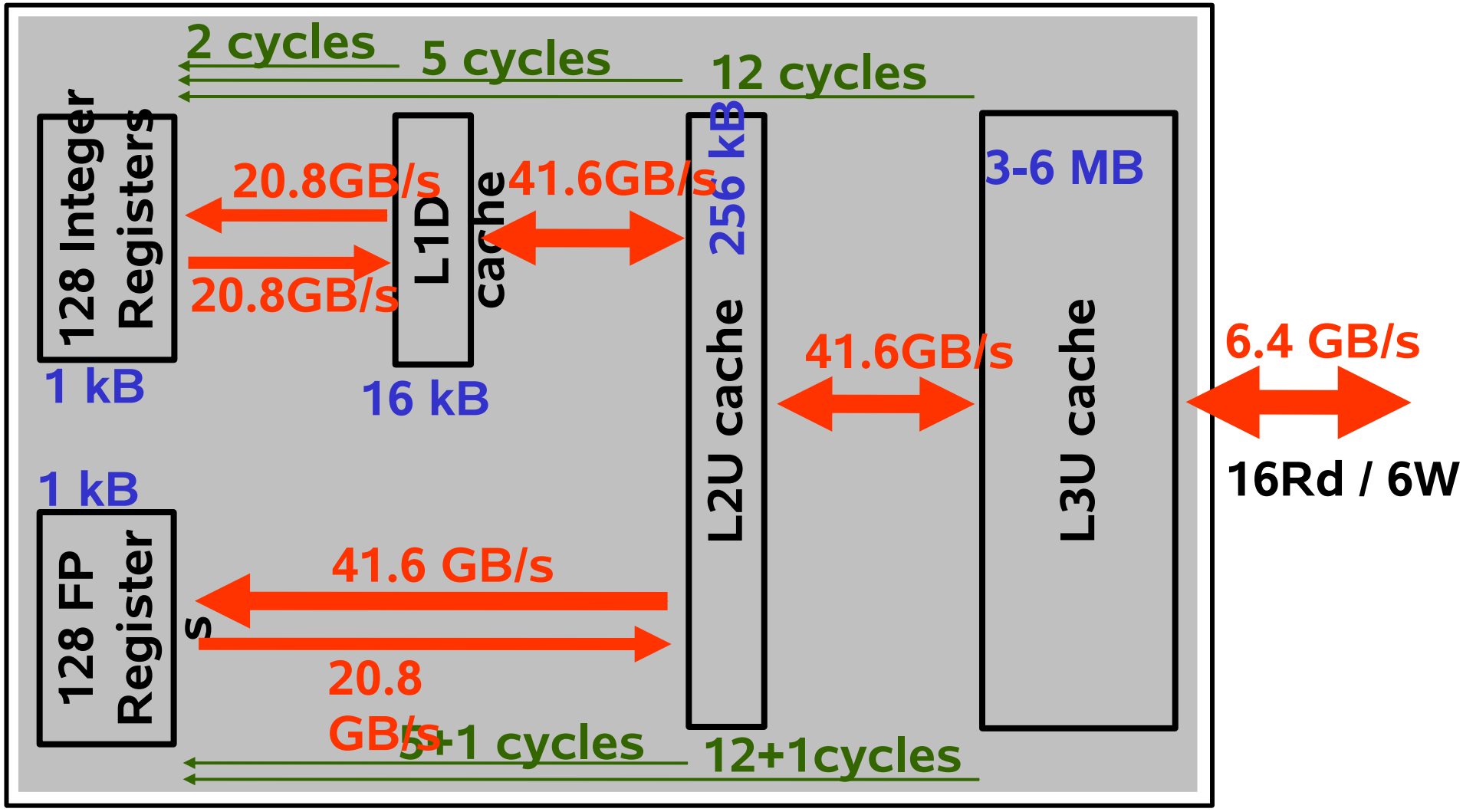
Similarly integer **, /, %(modulo)* are expanded through macros.

Latency will vary depending with compiler efficiency :

FP cycle	y = a+y	y = a*y	y = a+b*y	y = b+a/y	y = a/sqrt(y	y = sqrt(y	y =y/sqrt(y)
Single	4	4	4	28	36	43	36
Double	4	4	4	32	37	55	37

Int cycle	i = i + c	i = a*i	i = a+b*i	i = b+ a / i	i = b + a % i
Single	1	15	16	37	42
Double	1	15	16	56	61

Itanium™2 Data Flow (1.5 GHz)



Itanium™2 L1/L2 Data Cache

L1D is 16kByte, 64Byte/line, 4way, WriteThrough, GRegisters only:

- 1 cycle latency (2 for load, pointer chasing), no FP cached in L1D
- Store uses 8x8 bytes array. Updates L1D only if hit.
- 8 (unique) outstanding misses

L2U is 256kByte, 128Byte/line, 8way, WriteBack, NotRecentlyUsed

- 5,7,9.../6,8,10… latency for Int/FP
- 16 banks - 16bytes/bank (??? 256Byte stride/alignment ???)
- 16 (unique) outstanding misses
- L2 is not inclusive of L1D and L1I

Itanium™2 L3U Cache/Memory

L3U: 1.5/3MByte, 128Byte/line, 6/12way, WriteBack, LeastRecentlyUsed

- 12,16.../13,17… latency for Int/FP
- 16 (unique) read misses
- 6 write

Local/remote memory is accessed through SHub/NUMAflex:

Local latency	132 ns
Same brick / other node	180 ns
NL4 router	~50 ns
1 Meter cable	~10 ns


sgi®