

Cell Processor

Part A: Cell in Context

- Design a high performance processor
- How performance increase has been attained in the past
 - Higher frequency
 - Moore's Law
 - Architecture enhancements
- Why old tricks no longer work
 - Power wall
 - Memory wall
 - Architecture wall

Part B: Cell in Detail

- Introduce Cell
 - SPE
 - PPE
 - Memory controller, buses, IO
- Classifying Cell's topology
- Performance
 - Against Pentium 4, Xenon CPU
 - MPEG2 Decode
 - Optical flow
 - Games

Part C: Evaluation

- Reviewing Cell in terms of initial problems
- Reviewing Cell through P&H design guidelines
- Conclusion

Clock speed improvement

- Clock speed improvements from new fab processes
- 1989: 486 at 25MHz
- 2005: Pentium 4 at 3.6GHz

Improved Architecture

- Moore's Law:

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year."

-Gordon Moore, April 1965, Vol 38 Num 8, Electronics

- Implication: At the same price point, the number of transistors on a chip increases exponentially
- In 1975, Moore revised it to every 24 months
- Never said 18 months

Using Moore's Law

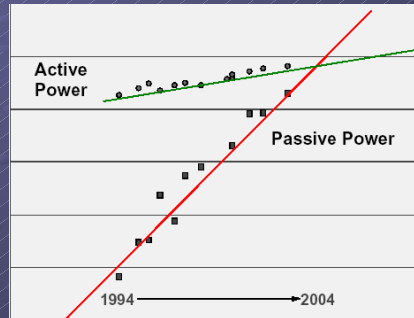
- Intel's Pentium:
 - 486: pipelining, FP unit
 - Pentium: superscalar, MMX
 - Pentium Pro: SSE, OOE
 - Pentium 4: Hyper-threading, large cache
- Mainly ILP exploitation
 - Theme of CS4211 so far: increasing ILP further is hard!

Why old tricks no longer work

- IBM's Peter Hofstee identifies three "walls"
 - Power wall
 - Frequency wall
 - Memory wall
- I propose one more:
 - Idea wall – "Out of micro-architectural ideas!"

Power Wall

- Gelsinger's Law: 40% performance increase (from design) every time transistor count doubles
- Corollary: Decreasing efficiency
- Too many transistors not doing real work
- Need to boost efficiency



Frequency Wall

- Pentium 4 2001 – 2003
- 1.5GHz to 3.06GHz
- 2003 – now: 3.6GHz
- Problem: power
 - October 2004: 4GHz Pentium cancelled
 - IDF 2005: Intel declares new direction as “multi-core”

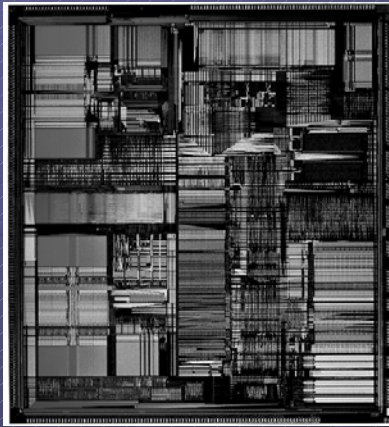
Memory Wall

- Memory wall is a *huge* problem
 - 80486DX
 - CPU and Memory runs at 33MHz
 - Memory latency at about tenths of cycles
 - Pentium 4
 - CPU ~3GHz, memory ~200MHz (DDR)
 - Memory latency at many hundreds of cycles
 - “Approach a thousand cycles in multi-GHz SMP systems” –Hofstee

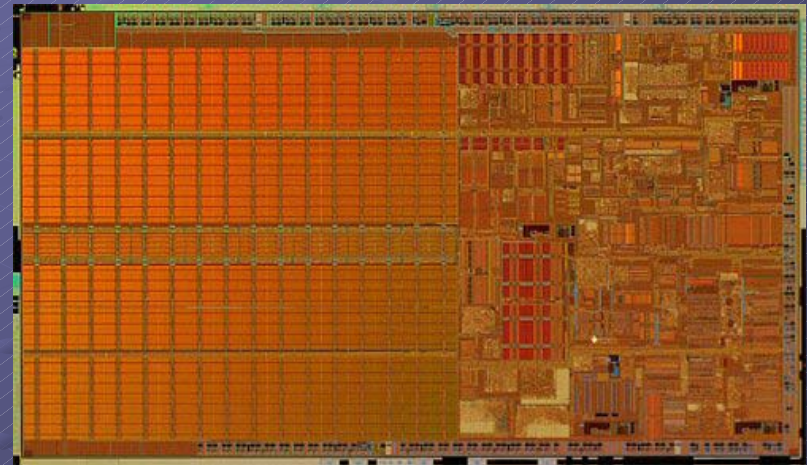
Idea Wall

- OOE/Speculation/Superscalar exhausted
 - “Pipelining was the last major win” -Hofstee

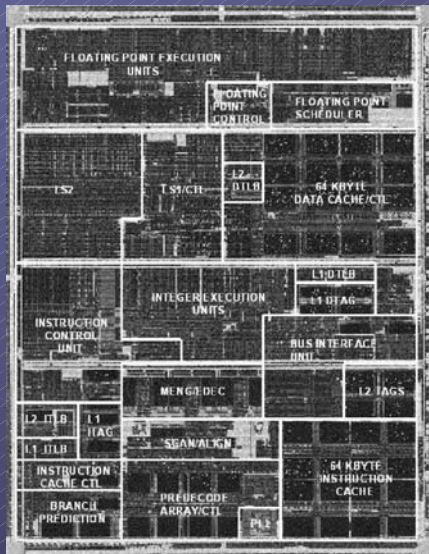
Original Pentium



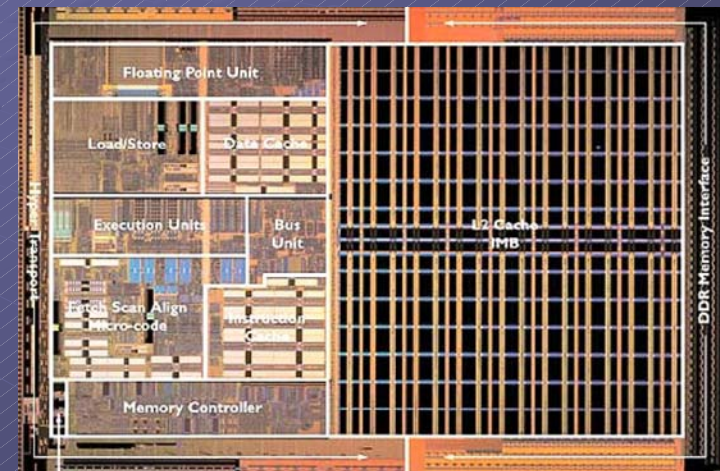
Pentium M (Dothan)

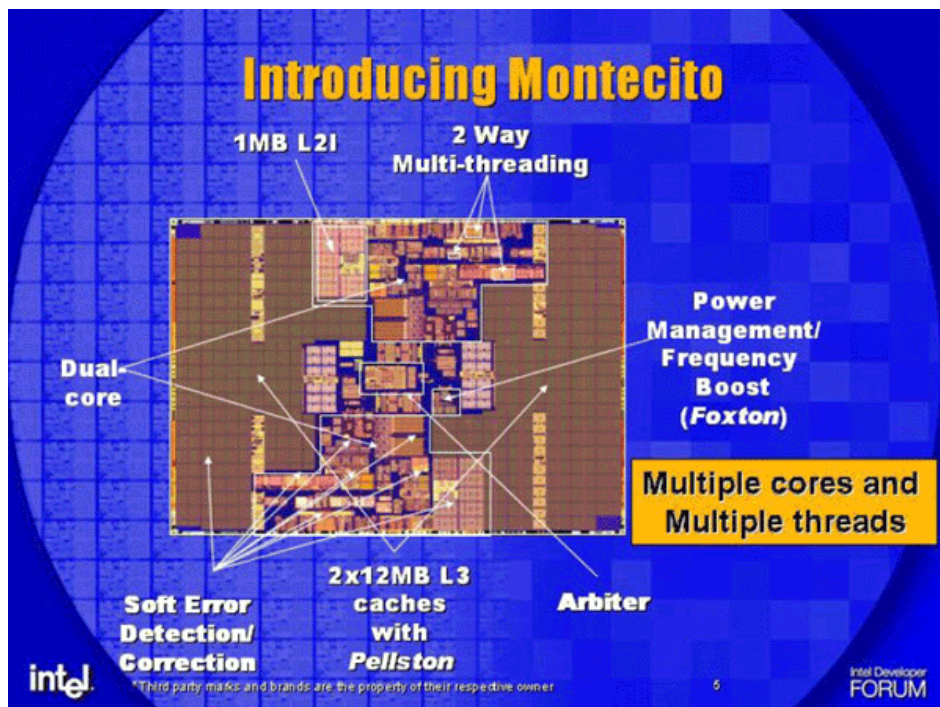


AMD Athlon (K7)



AMD Opteron (K8)





Idea Wall

- Lacking new architectural breakthroughs
 - “A crisis of ideas” – CS3211 Lecture 2004
- Indefinite increase in cache is not a solution
 - Diminishing returns

A summary of non-solutions

- More complex OOE, prefetching, execution units
 - Even if good returns, passive power dominates
- Bigger cache
 - Diminishing returns
- Higher frequency
 - Not even Intel believes it anymore

STI's solution: Cell

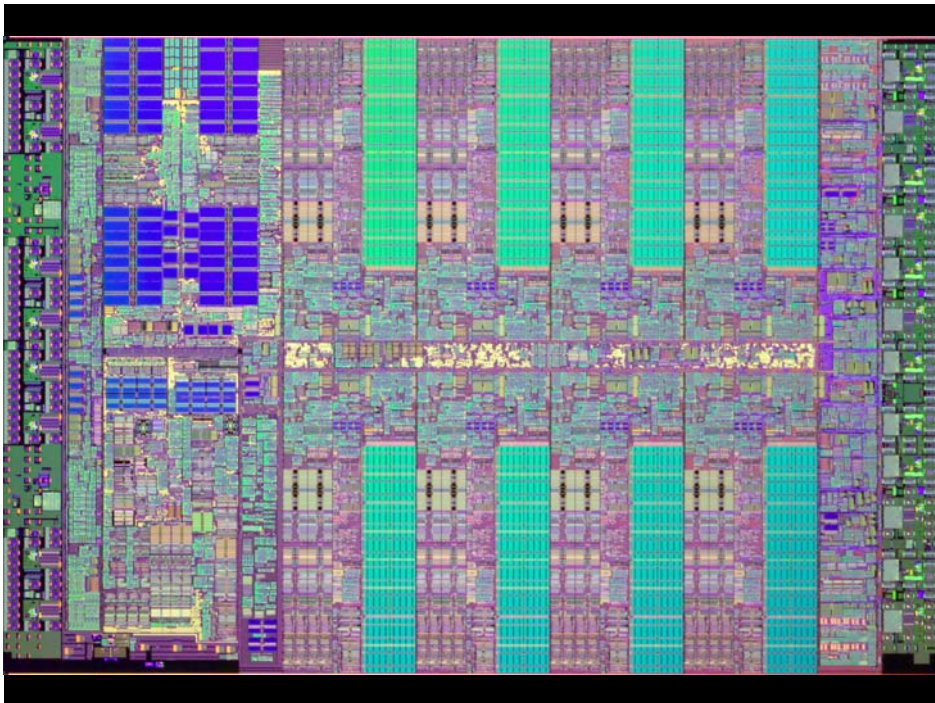
- STI: Sony, Toshiba and IBM
- Started in 2001 by Ken Kutaragi of Sony Computer Entertainment
- \$400M, 5 years, 300 engineers
- Attacks
 - Memory wall
 - Power wall
 - Idea wall

What's Cell designed for?

- Media applications (SIMD nature)
- Very high floating point performance
- High bandwidth
- Good performance to power ratio
- Characteristics:
 - Streaming nature
 - Scalable architecture
- Target applications:
 - Playstation 3
 - HDTV
 - High performance embedded

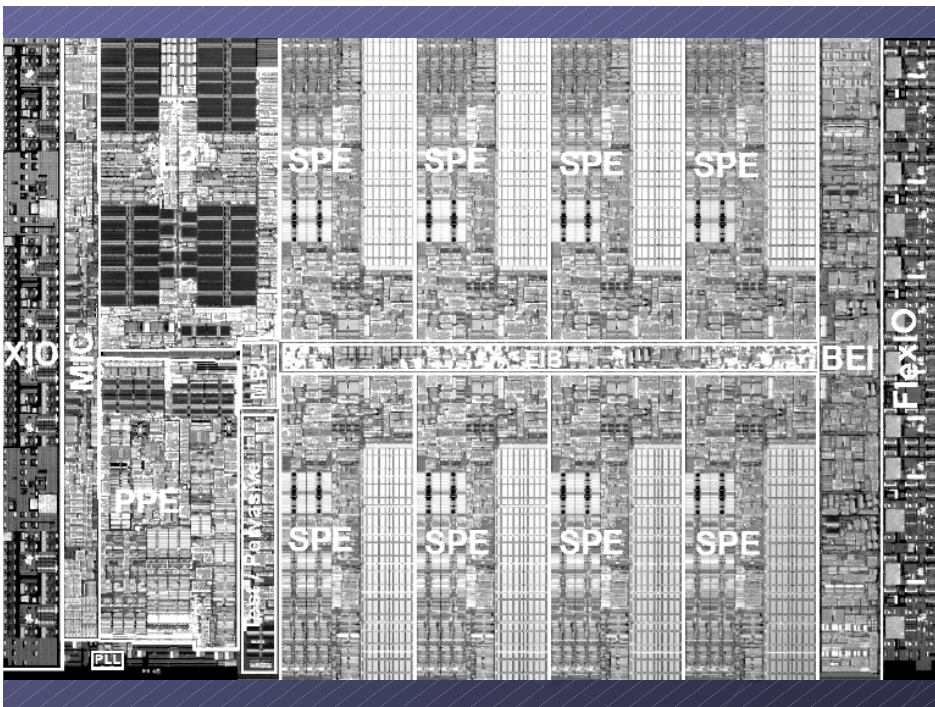
Inspirations

- Playstation 2 CPU: Emotion Engine
 - 2 Vector Units
- IBM BlueGene
 - Scientific computing
 - BlueGene/L currently #1 supercomputer
 - 32,768 processors
 - 91 Teraflops peak
- Cray Supercomputers
 - Little cache, high bandwidth



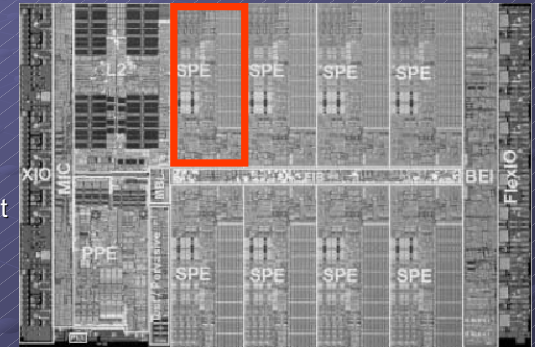
Key figures

- Fundamentals
 - 234 million transistors
 - 90nm SOI Process
 - 221mm² die
 - 4GHz clock speed
- Architecture
 - 64-bit Address space, 128-bit SIMD
 - 9 CPU cores on chip
 - Integrated memory controller
 - 100 GB/s total bandwidth (memory, IO)



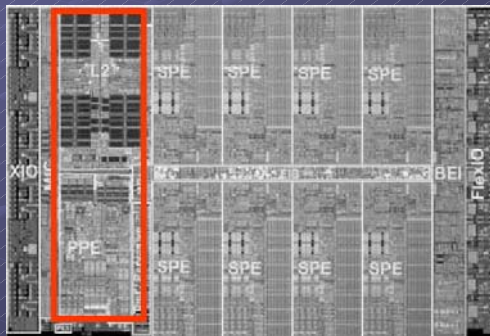
Synergistic Processing Element (SPE)

- Heart of the Cell architecture
- Scalable resource
- 128-bit SIMD processors
 - Executes any 128-bit combination in one clock:
 - 16 chars
 - 8 shorts
 - 4 Integers or floats
 - 2 Doubles



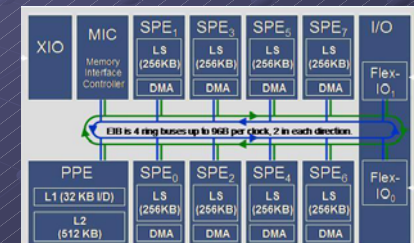
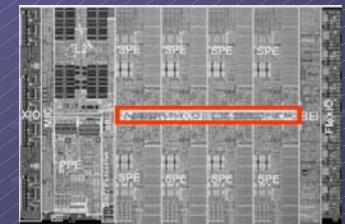
Power Processing Element (PPE)

- 64-bit Power Architecture
 - Designed for OS and general purpose computing
 - In order, dual issue
 - Hardware multi-threading for OS virtualisation
- Conventional memory hierarchy
 - 32kb Instruction cache
 - 32kB Data cache
 - 512kB L2



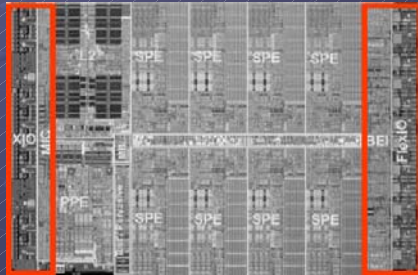
Element Interconnect Bus (EIB)

- Ties up all the elements on the chip
 - PPE
 - SPE
 - L2 cache
 - Memory controller
 - IO
- 4 x 16 Byte data rings
- 96 Bytes a cycle peak
- Over 100 outstanding requests



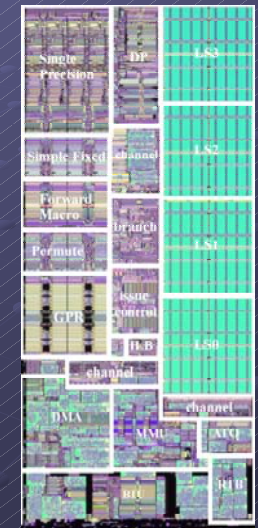
Memory and IO

- Licensed RAMBUS technology
 - XDR Memory
 - High frequency and narrow
 - 64-bit bus @ 3.2GHz
 - DDR can be supported via external bridge
- XIO <-> MIC talks to main memory
 - 25.6GB/s
- BEI <-> FlexIO talks to IO
 - Two configurable interfaces
 - For GPU, more Cells, IO bridge etc.
 - 76.8GB/s total



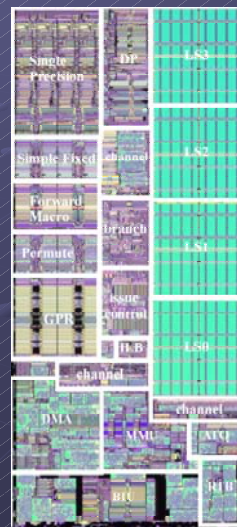
Synergistic Processing Element

- Not co-processor, but entire, autonomous computer
 - Datapath
 - Control
 - Memory
 - Input
 - Output
- 128 x 128bit Registers
- 256kB Local Store (LS)
 - Unified instructions and data store
- DMA and MMU units
- "Memory anaemic vector computer"



SPE Pipeline

- Dual issue, statically scheduled, in order SIMD pipeline
- The key is in what they've taken out:
 - No Tomasulo
 - No prefetching
 - No speculation
 - No branch prediction tables
 - No cache logic
- In exchange for:
 - Bigger local store & registers
 - Wider execution unit
 - Faster clock
 - Smaller footprint (more cores)
 - Better efficiency (Attacks Power Wall)
- Bottom line: More hardware devoted to work rather than overhead



SPE Performance

- Built around 32-bit Floats
- 4 x 32-bit FLOPS / cycle
- 8 FMACS / cycle
- At 4GHz = 4 * 8 = 32GFlops
- Integer 4 x 32-bit Int / cycle
- DP is 10x slower

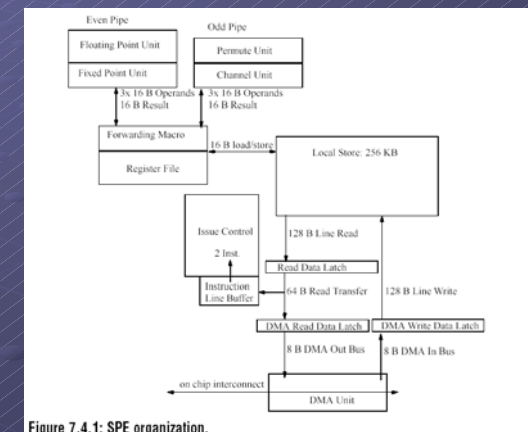
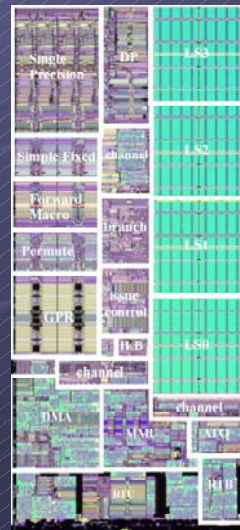


Figure 7.4.1: SPE organization.

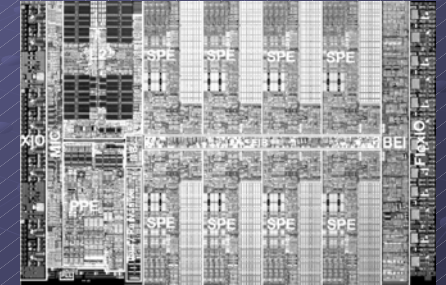
SPE Bandwidth & Latency

- Register file
 - 6 read ports
 - 2 write ports
 - 2 cycle latency
- Local store
 - 6 cycle read
 - 4 cycle write
 - 16B / cycle
- Compare with a PPC 970:
 - 11 cycle L2 cache
- 128 B / cycle DMA



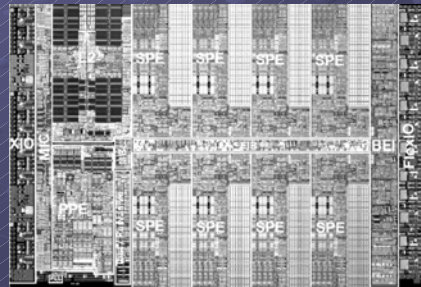
Classifying Cell

- It's been called all sorts of things
 - Processor with 8 co-processors
 - DSP
 - Stream processor
 - CPU/GPU hybrid
 - Supercomputer
- Bottom line: On-chip multiprocessor



Classifying Cell

- Using P&H multiprocessor dichotomy:
 - Address space
 - UMA
 - NUMA
 - Memory location
 - Centralised (dancehall)
 - Distributed
 - Connection method
 - Bus
 - Network



Classifying Cell as a Multiprocessor

- In terms of address space
 - Single chip
 - UMA + Message passing
 - Processors not identical, hence not SMP
 - Multi-chip
 - NUMA + Message passing

Arrangement	Memory Address		
	Shared:		Private
	UMA	NUMA	Message Passing
One Cell Chip	All main memory access have same delay		SPE to SPE access
Cell Network		Main memory access between nodes	SPE to SPE access

Classifying Cell as a Multiprocessor

In terms of memory location

- Single chip
 - Centralised main memory
 - Distributed local store
- Multi-chip
 - Distributed main memory
 - Distributed local store

Arrangement	Memory Location	
	Distributed	Centralised
One Cell Chip	SPE local store	Main memory
Cell Network	Distributed main memory and LS	

Classifying Cell as a Multiprocessor

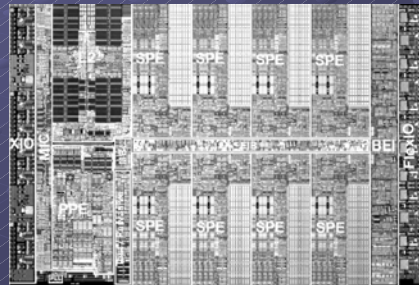
In terms of connection

- Single chip
 - EIB Bus ties everything together
- Multi-chip
 - External network IO

Arrangement	Connection	
	Bus	Network
One Cell Chip	Main memory, on-chip SPE	
Cell Network		Between 2 or more Cells

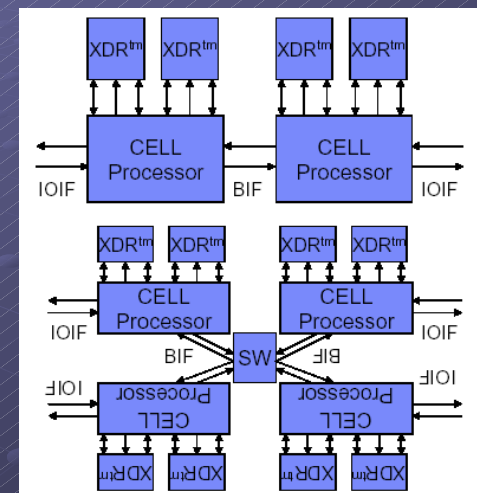
Cell's typology

- Main memory is 9 way UMA
 - 1 PPE + 8 SPEs
- Local store of SPE is private memory with message passing
- Typology: Uniform access shared memory + message passing private memory
 - Single Cell resembles a cluster of memory anaemic vector computers



Cell's typology

- NUMA
 - Multiple Cell networks resembles homogenous grid



Performance

- No real world benchmarks released
- Evaluate peak performance
 - Each SPE capable of 32GFlops
 - 8 SPE = 256GFlops
 - “One CELL has a capacity to have 1TFLOPS performance” – Ken Katuragi
 - 4 Cells on chip is just a matter of time

Cell vs. Pentium 4

- Pentium 4 Dual Core @ 3.5GHz
 - Roughly the same process technology as Cell
 - 250 Million transistors
 - 90nm
 - Single core = 3.5×4 (SSE) = 14 GFLOPS
 - Dual core = 28 GFLOPS
- Cell @ 3.5GHz = 224GFLOPS
- About 10x higher peak performance

* Excludes P4 FPU and Cell PPE

Cell vs. Xenon

- Xenon
 - 3 Cores (same as PPE)
 - Each more has 1 FPU and 1 VMX Unit
 - 10 Flops / cycle
 - 2 Flops / cycle from FPU
 - 8 Flops / cycle from SIMD
 - At 3.2Ghz = 96GFlops
 - Cell at 3.2GHz = 205GFlops
- About 2x higher peak performance
 - Not counting Cell's PPE

Applications for Cell

- HDTV
- Toshiba demonstrated Cell decoding 48 SDTV streams
- Sony demonstrated Cell decoding 12 HD streams
- Read / decode
- Resize
- Final 1920 x 1080
- Six SPEs used



Applications for Cell

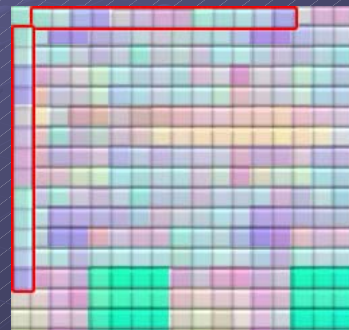
- Optical Flow Algorithm for path finding
- Input:
 - 567 x 378 @ 27.4FPS
 - About 6MPixels / second
 - 8-bit greyscale pixels
- Algorithm:
 - 5 stage pre-processing
 - Gauss-Seidel iteration
 - 6000 passes per fame!
- Currently work is being done on a FPGA solution

Applications for Cell

- A good match for Cell
 - Float intensive
 - SIMD
 - SPE friendly
 - Small instruction size
 - Highly iterative
 - Good for reusing local store

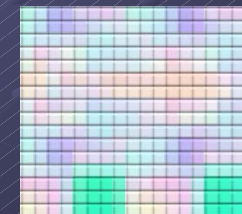
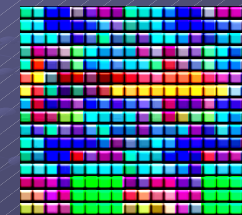
Applications for Cell

- Optical Flow Pre-processing
- 1. Smooth
 - 13 element mask
 - Multiply each element by a factor
 - Sum results
 - Assuming 16 element mask
 - 16 FMAC per pixel
 - 16 * 6MPixels = 96M FMACs / sec
 - 192FMACs/s for both X and Y axis



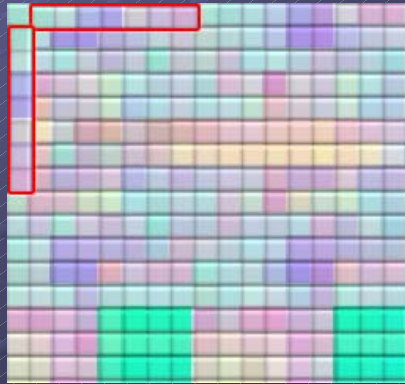
Applications for Cell

- Optical Flow Pre-processing
- 2. Temporal Gradient
 - Calculate difference between raw and smoothed frame
 - 1 Subtract per pixel
 - 6 MPixels
 - 6 MFLOPS
 - Produces 1 frame (Ft)



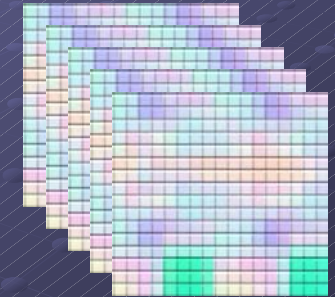
Applications for Cell

- Optical Flow Pre-processing
- 3. Spatial Gradient
 - Same calculation as smoothing
 - 7 element mask, assume 8
 - 8 FMACs / pixel
 - 48M FMACs / sec
 - 96M FMACS to produce 2 frames (F_x , F_y)



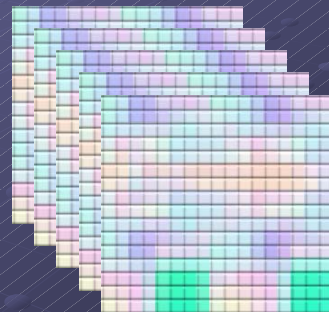
Applications for Cell

- Optical Flow Pre-processing
- 4. Compute 5 different frames
 - $F_{xx} = F_x * F_x$
 - $F_{yy} = F_y * F_y$
 - $F_{xy} = F_x * F_y$
 - $F_{tx} = F_t * F_x$
 - $F_{ty} = F_t * F_y$
- 1 FLOP / pixel, 6MFlops /sec
- 30MFlops / sec for 5 frames



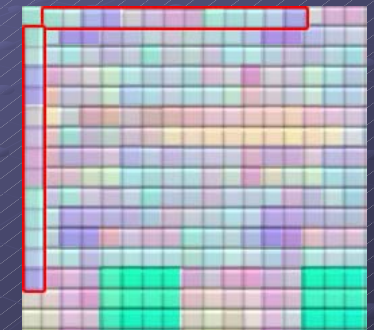
Applications for Cell

- Optical Flow Pre-processing
- 5. Final smoothing
 - 11 element mask, assume 12
 - Smooth all five output frames
- 12 FMACs / pixel
- 72M FMACs /sec
- 360M FMACs / sec for 5 frames



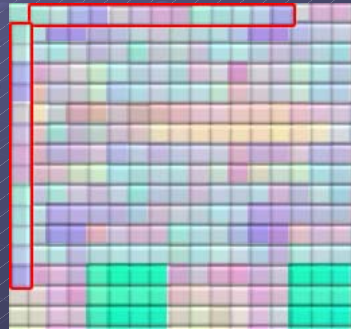
Applications for Cell

- Optical Flow Pre-processing
- 5. Pre-processing finished
 - $192 + 6 + 96 + 30 + 360 = 684\text{MFLOPs/s}$
- Easily done by 1 SPE



Applications for Cell

- Optical Flow
 - Pre-processing over
 - Final stage is the *real* work
- Gauss-Seidel iteration
- 13 FLOPs / pixel
- 78 MFLOPs / sec
- For 6000 iterations
 - 468GFLOPs
- Will need two Cells!
 - With optimisation, possible with 1
 - Use 16-bit for pixels
 - Alt. algorithm (successive over relaxation)



Evaluating Cell

- Initial problems
 - Power wall
 - Frequency wall
 - Memory wall
 - Idea wall

Evaluating Cell

- Power wall and frequency wall
 - Due to leakage and poor utilisation
- Removed
 - Tomasulo
 - Prefetching
 - Speculation
 - Branch prediction tables
 - Cache logic
- In exchange for:
 - Bigger local store & registers
 - Wider execution unit
 - Faster clock
- Result:
 - Better efficiency

Evaluating Cell

- Memory Wall
 - Up to 1000 cycle memory latency
 - A die full of cache
- Cell
 - 6 cycle read to local store
 - Twice as fast as L2 cache
 - Modest cache

Evaluating Cell

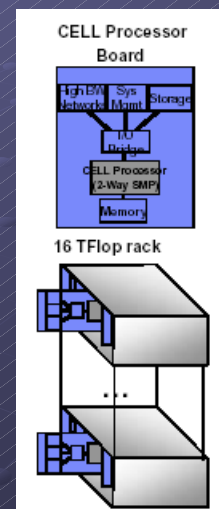


Evaluating Cell

- Idea wall
 - Don't know what to do with extra transistors
 - Lack of useful micro-architectural enhancements
- Cell
 - High scalability
 - More SPEs more Cell
 - More Cells per chip
 - Smaller, faster Cells, more of them across network

Evaluating Cell

- 2 Cells per die at 65nm
 - Glueless SMP
- 32 Blade rack yields 16 Teraflops
- 1 Peta-flop in 64 racks



Patterson & Hennessy Guidelines

- 1. Simplicity favours regularity
 - Each SPE is designed to be as simple as a SIMD core can be
 - Cell is a collection of simple and identical SPEs
- 2. Smaller is faster
 - SPE is small
 - Runs in access of 5GHz by itself

Patterson & Hennessy Guidelines

- 3. Good design demands good compromises
 - Each SPE has only 256kB of memory
 - Not good for many applications but works well for decoders and stream kernels
- 4. Make the common case fast
 - Common memory access use to be L2 cache at around tens of cycles
 - SPE local store has 6 cycle read
 - Very common graphics operation is multiply-accumulate. SPE supports MAC in one cycle

Finally...

- Playstation 3 announced yesterday
- Cell at 3.2GHz
- 7 SPEs
- NVIDIA GPU
- 512MB total memory



Finally...

- Currently 80 Million PS2s sold world wide
- 80 Million PS3s will yield 14 Million Teraflops



Conclusion

- Cell addresses many of the big problems
 - Power wall
 - Memory wall
 - Frequency wall
 - Idea wall
- Cell is scalable
 - Suitable for many platforms
- To the get most out of Cell
 - New programming models

References

- H. Hofstee (2005) *Power Efficient Processor Architecture and The Cell Processor*, Proceedings of the 11th Int'l Symposium on High-Performance Computer Architecture (HPCA-11 2005)
- B. Flachs et al. (2005) *A Streaming Processor Unit for a Cell Processor*, 7.4 Multimedia Processing, ISSCC2005
- Paul Zimmons (2003) *Cell Architecture*
- Mark DeLoura (2005) *Cell - A new platform for digital entertainment*, Game Developer's Conference 2005, SCEA