The Internet Streaming SIMD Extensions

Shreekant (Ticky) Thakkar

Intel Corp.

Outline

- Goals
- Architecture
- Instruction Set Arch./Benefits
- Status
- Conclusion

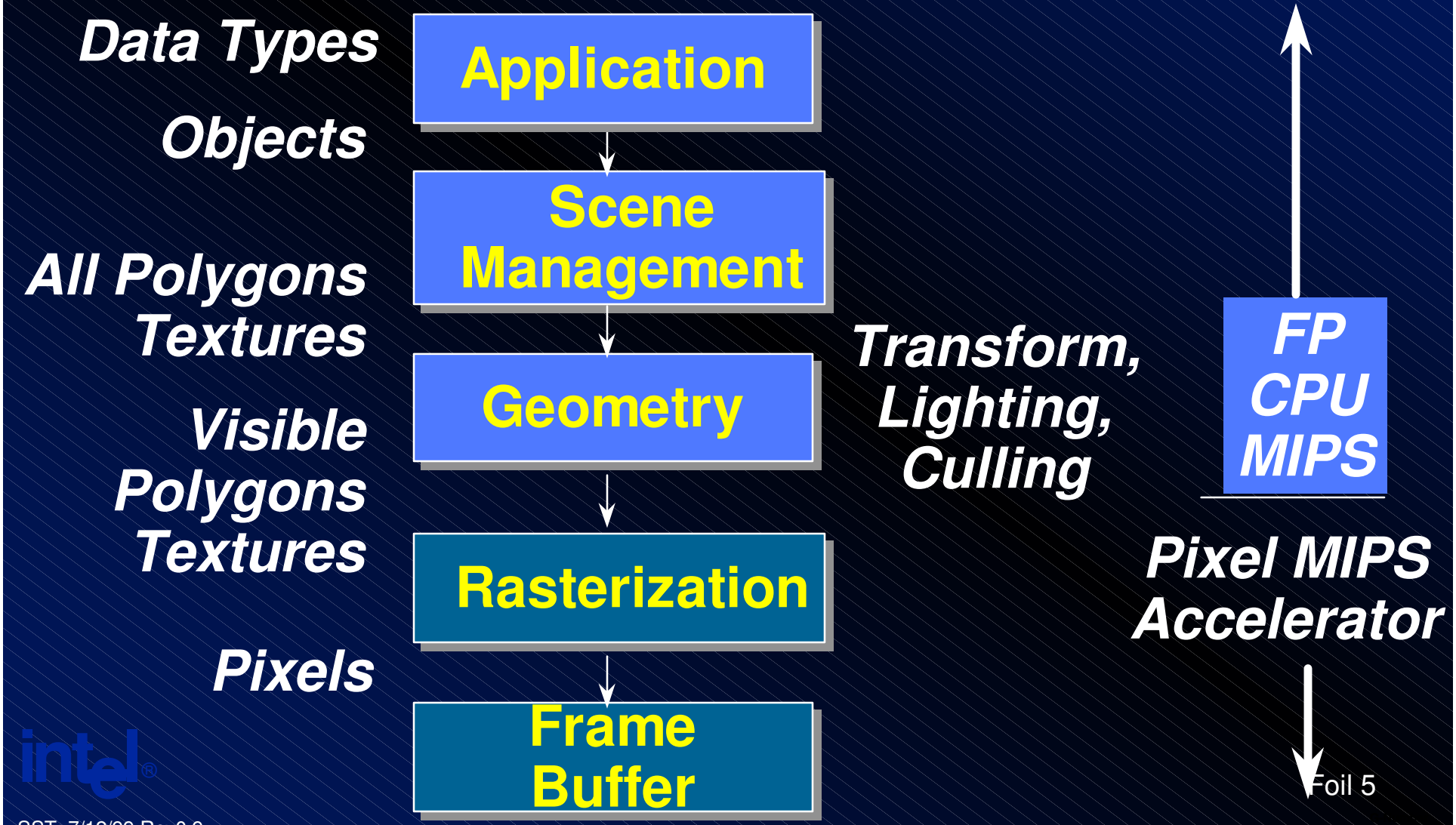
Goals

- Make PC platform of choice for 3D
- Visual perception change requires > 2x perf.
- Accelerate/enable other multimedia apps. (Physics, Video, Speech, Image)
- Multiple designs must be able to implement new architecture extension
- Biz model - ubiquitous availability & perf. improves w/each process/uarch

Outline

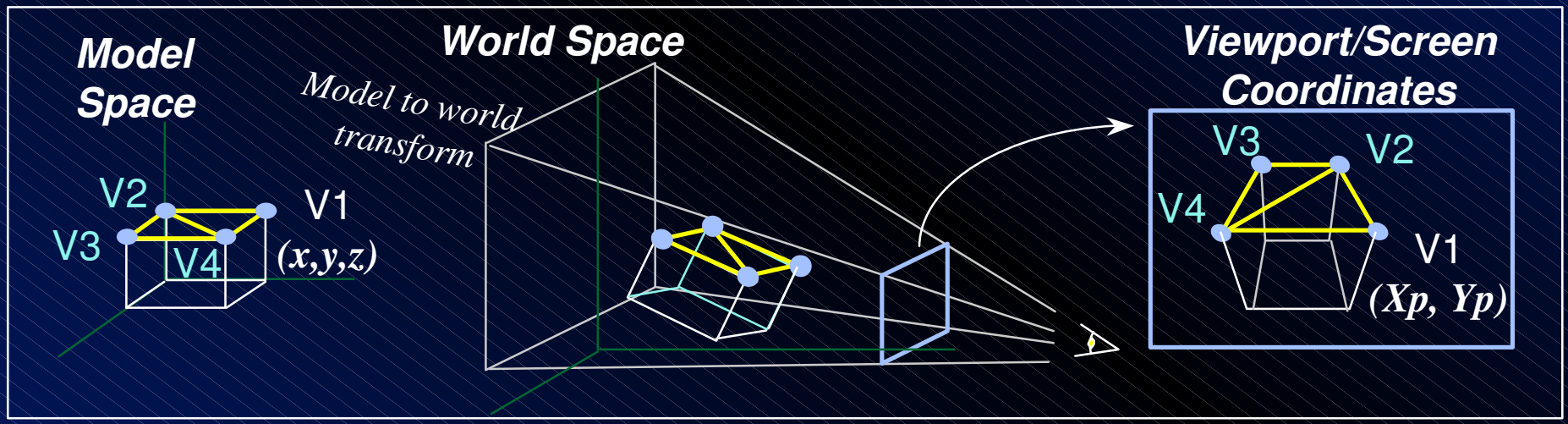
- Goals
- **Architecture**
- Instruction Set Arch./Benefits
- Status
- Conclusion

3D Graphics Pipeline



3D Geometry is Data Parallel

- Compute x, y, z in parallel per vertex
- Compute multiple vertices in parallel

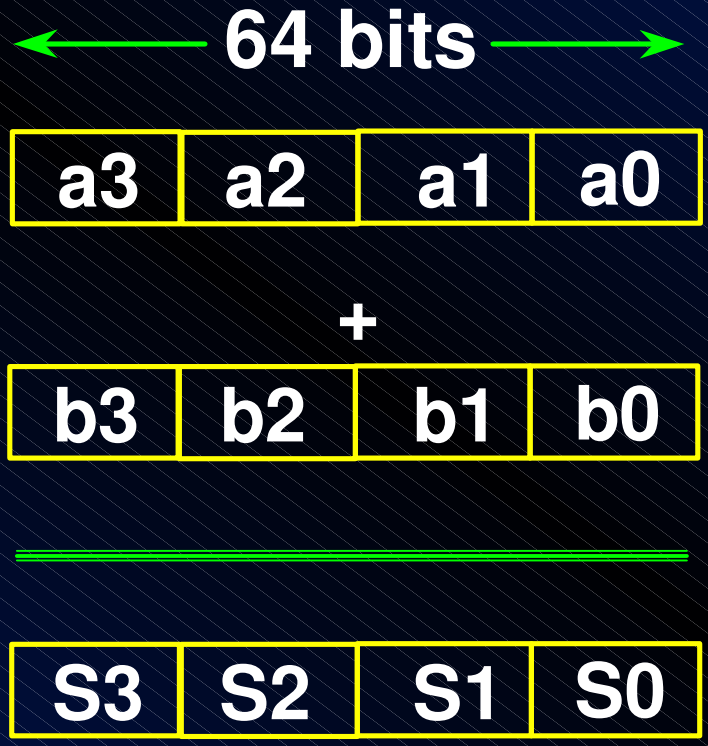


SIMD FP is best option to deliver > 2x perf. gain

Learning from MMX™ Tech.

- Overlapped registers impact app. usage
  - –new flat registers, simultaneous use
- Balance bandwidth to support execution
  - –concurrency through prefetch/streaming
- Better conditional flow support
  - –add rich set of compare, mask move inst.
- Get data quickly into packed format
  - –shuffle, unpack/pack, movl/hps

MMX™ Tech. added SIMD Int.

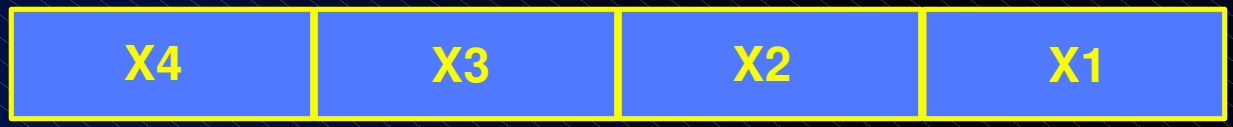


where:
sn = an+bn

Packed Integer (32,16, 8 bits) Op./instruction

ISSE Adds SIMD-FP

- Packed & Scalar FP Instructions Operate on Packed 4 Single Precision Numbers
  - Packed Instructions Work on 4 Numbers
  - Scalar Instructions Work on LS Number



New SIMD-FP ISA

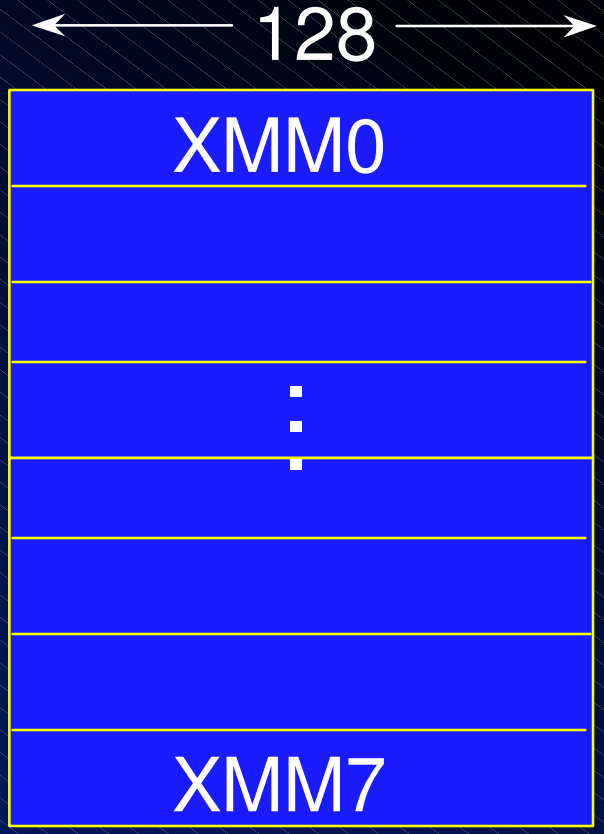
- IEEE 754 Compatible FP Arithmetic
  - Masked Support Same x87
  - Unmasked Support Requires New Handlers
- Flush to Zero Mode for 3D Graphics
  - Underflows Are Flushed (rounded) to Zero

New State

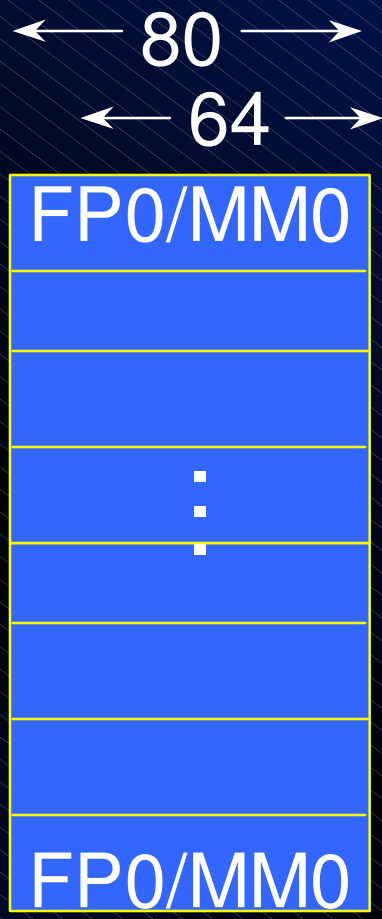
- 8 x 128 bit Flat Registers
  - Complete New State in IA-32 Architecture
  - MMX™ Technology/IA-FP and Streaming SIMD Extensions Synergy:
    - Instructions Scheduled For Simultaneous Execution
    - Conversion Support Between Integer and FP
- New Status/Control Word
  - Combined Control/Status Double Word (32 bit)
  - New LD/ST Instructions

New ISSE Flat Registers

ISSE Registers
(Scalar/packed SIMD-SP)



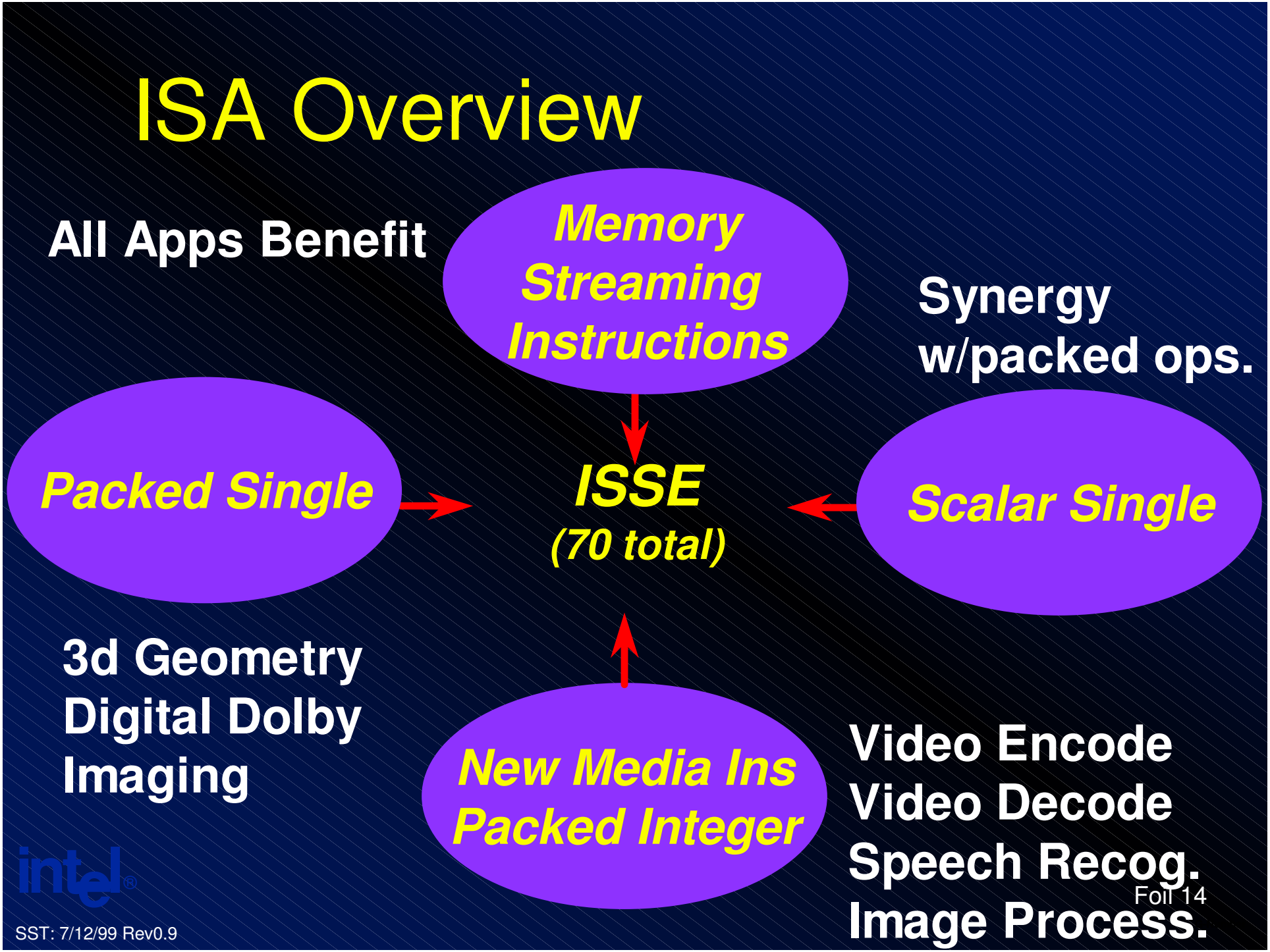
MMX/x87 Registers
(64 bit Integer, x87 data)



Outline

- Goals
- Architecture
- **Instruction Set Arch./Benefits**
- Status
- Conclusion

ISA Overview



Expected Performance Gains

Applications	*Expected*
3D Geometry Kernel	2x (2.25x ~ 4MVx/s)
Collision Detect. Lib	40% (40%)
DVD Decode	6% (7%)
MPEG-2 Encode	Real Time @ 30fps (500/100 – 30%)
Speech Recognition	22% App. Gain

Measured results show we exceeded expectations

Packed/Scalar SIMD-FP

Applications	*Expected*
3D Geometry	2x (2.25x ~ 4MV/s)
Collision Detect. Lib	40% (40%)

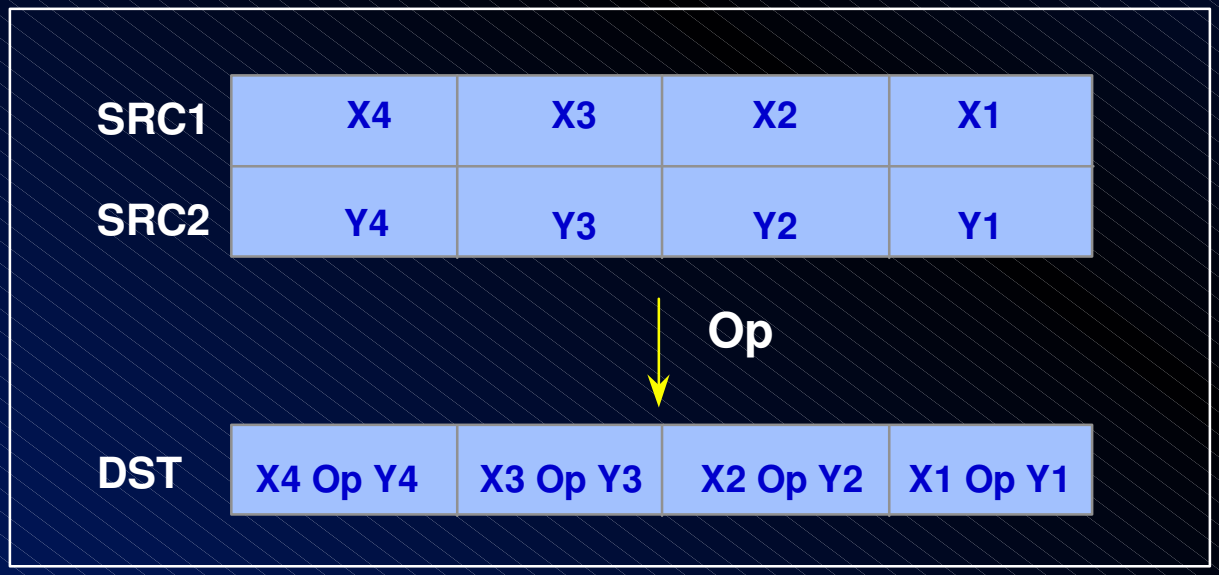
Arithmetic: ADD, SUB, MUL, DIV, MAX, MIN, RCP, RSQRT, SQRT,
Logic: AND, ANDN, OR, XOR,
Compare: CMP, MAX, MIN, COMI,UCOMI
Data Movement: MOV (aligned), MOVU, (unaligned),MOVLPS, MOVHPS, MOVMSK, SHUF, UNPCKH, UNPCKL

Computation Instructions

- FP Arithmetic (Packed/Scalar)
  - Vertical: ADD, SUB, MUL, DIV, MIN, MAX
  - Square Root: SQRT
  - Lookup: RCP, RSQRT
- Logic (Packed)
  - AND, ANDN, OR, XOR
  - Used for Masking

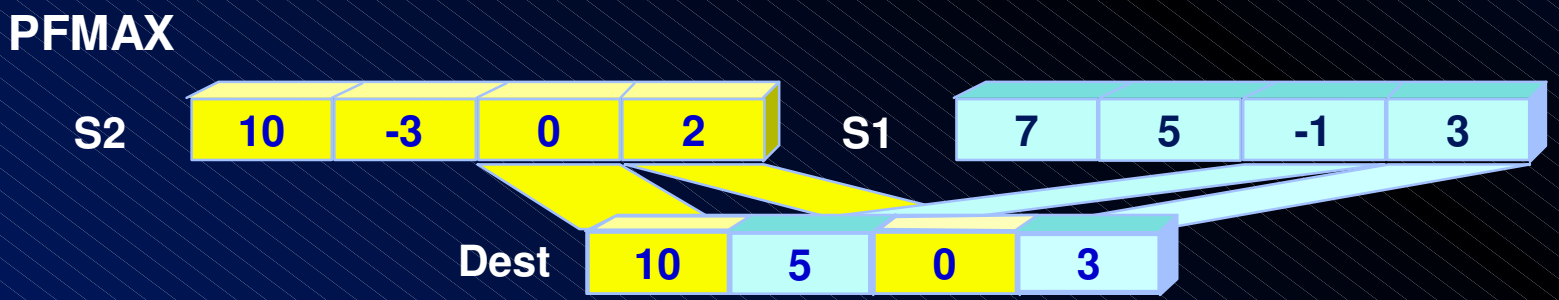
ADD/SUB/MUL/DIV

- Operate Vertically On:
  - Packed: 4 x SP Ops in Parallel (w/PS Suffix)
  - Scalar: 1 x SP Op (w/SS Suffix)



MINPS/MAXPS

- Two Comparison Instructions Which Return Larger/Smaller of 2 Sources:
  - Packed: Return Min/Max of 4 Pair of Ops.
  - Scalar: Return Min/Max of Lowest Ops.

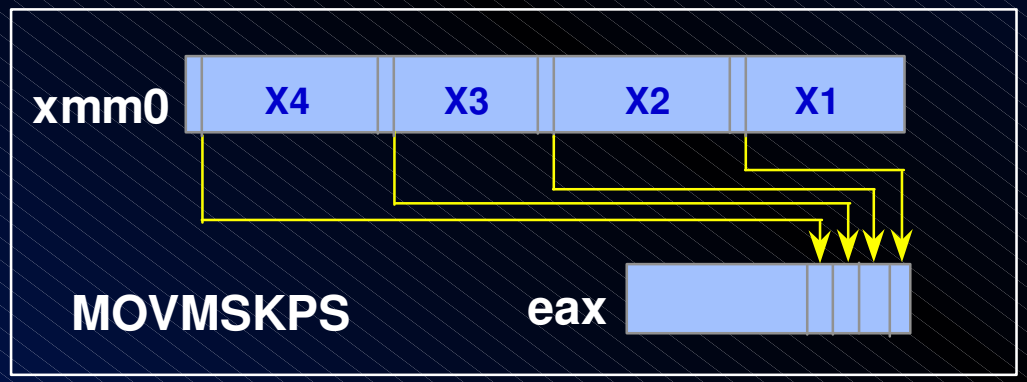


Comparison Instructions

- Compare
  - Pack Compare: CMP
    - Immediate Op Gives Following Relationships:
      - EQ, LT, LE, UNORD, NEQ, NLT, NLE
    - Does Not Update E-flags
    - Mask Generated
  - Scalar (IEEE) Compare: PFCOMI, PFUCOMI
    - Updates E-flags

Move Mask Instruction

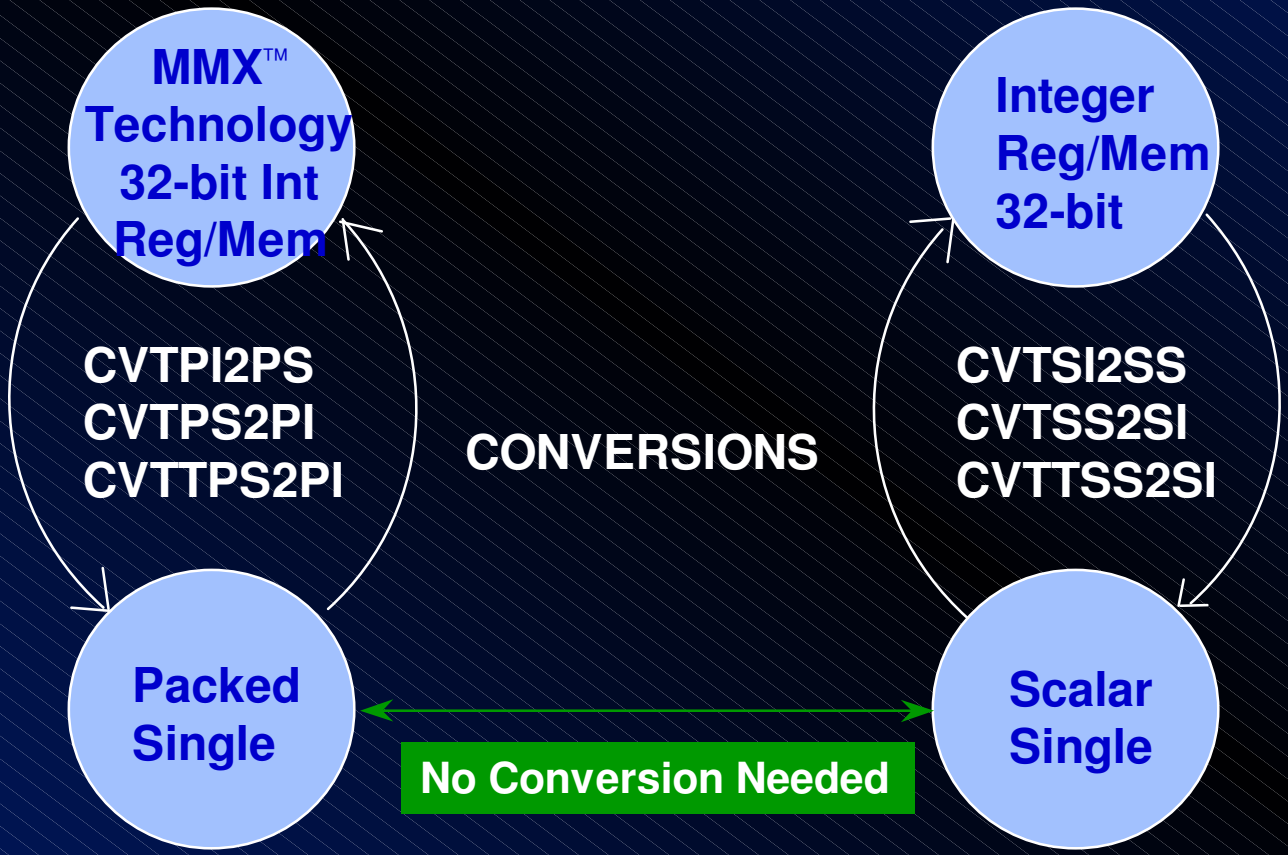
- MOVMSKPS
  - Moves Most Significant bit of Each Pack Compare Result to Integer Register



Conversion Instructions

- Scalar Single ↔ 32-bit Signed Integer
- Two LS Packed Single ↔ Two 32-bit Signed Integer (MMX™ Technology)

CONVERSIONS



Data Movement Instructions

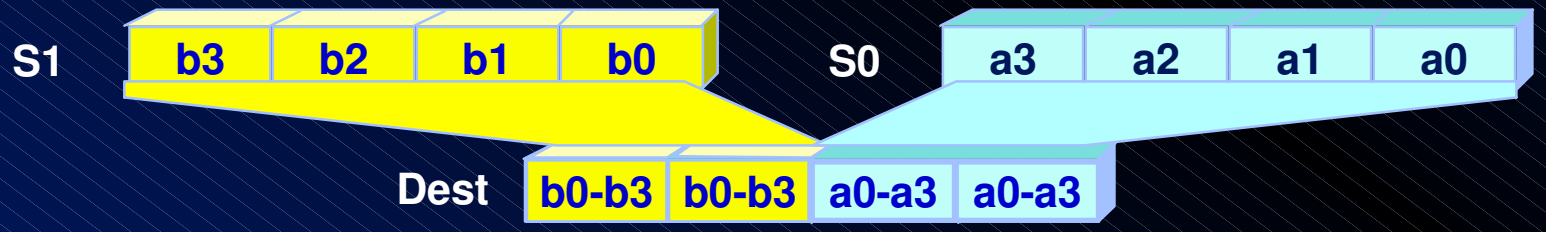
- MOVPS/MOVEUPS:
  - 128-bit Aligned/Unaligned Packed Load/Store
  - Unaligned Data With Aligned Load Will Fault
- MOVSS:
  - 32-bit Scalar Single Load/Store
  - Upper bits Are Cleared
- MOVLPS/MOVHPS
  - Load/Store Low/High 64-bit of 128-bit Packed

Data Swizzle Instructions

- These instructions enable programmer to get data organized in optimal SIMD form
  - Shuffle
  - Unpack
  - Special Loads/Stores

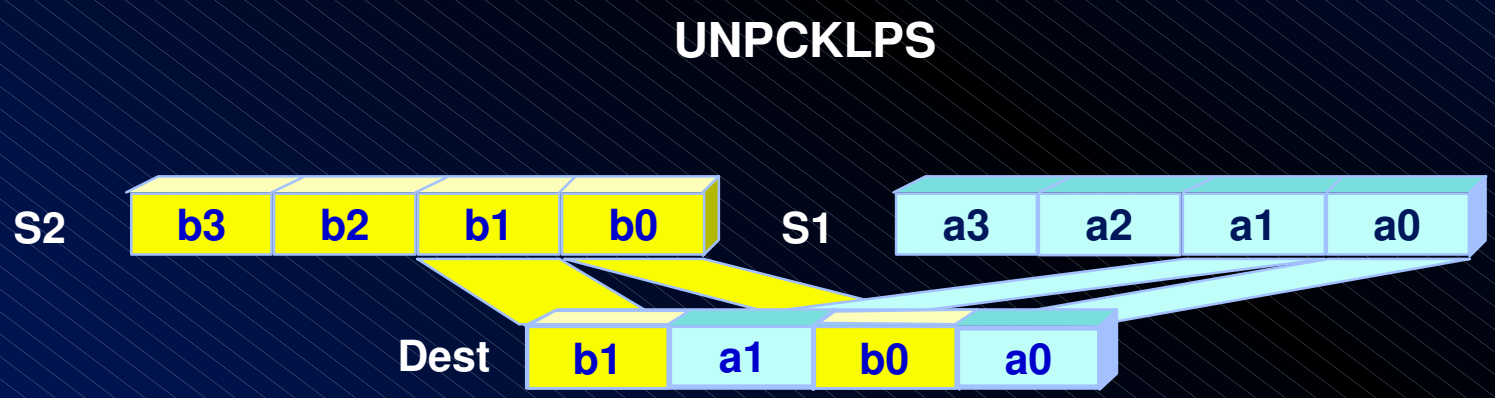
SHUFPS

- Moves 2 x SP FP Numbers From Each Source to the Destination Under Control of an 8-bit Mask
  - Full Shuffling If Both Sources Are Same
  - Performs: Rotate, Shift, Swap and Broadcast



UNPCKLPS/UNPCKHPS

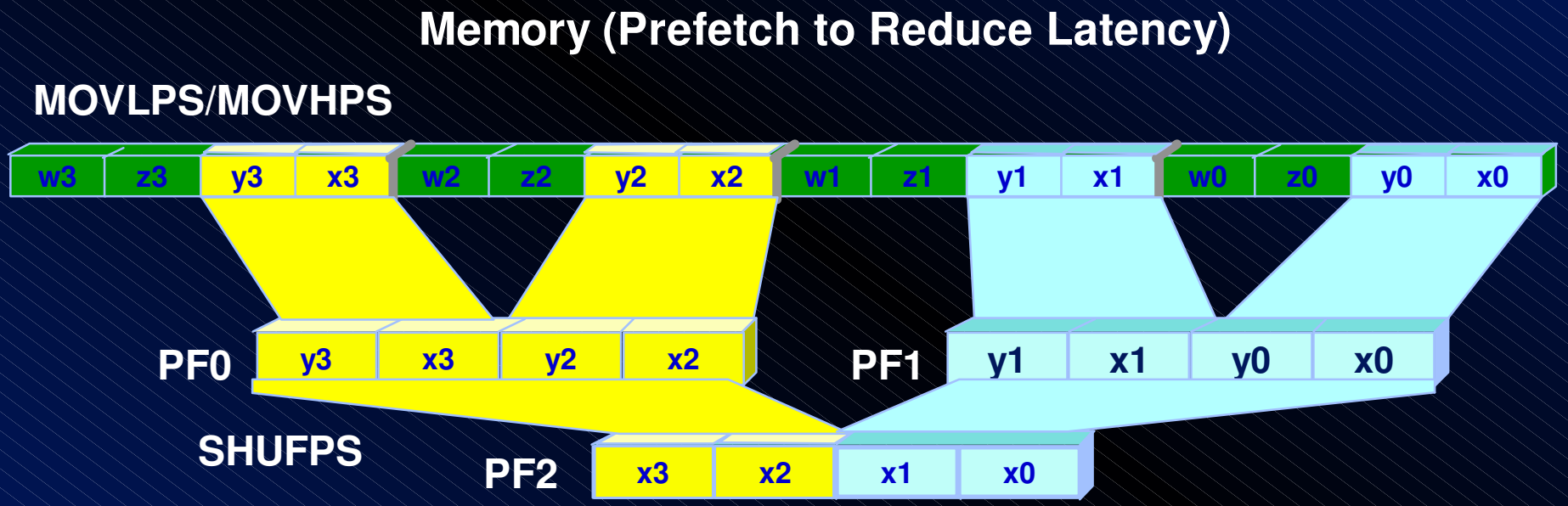
- Interleave Single-FP From Two Sources
- UNPCKHPS: Unpack High Single
- UNPCKLPS: Unpack Low Single



3D Data Swizzle

- Need to “SIMDify” Data to Produce 4 Results Instead of One Per Iteration
  - Use MOVLPS/MOVHPS (64b) and SHUFPS to Perform xyz->xxxx->xyz Packing/Unpacking
  - 20-25% Overhead Compared to Pre-Organized (xxxx) Data
  - Additional Computation Amortizes Overhead

3D Data Reorganization



New Media Instructions (Packed Integer)

Decode: PAVRGW; **Encode:** PSADW;
Speech: PMIN/PMAX, PINSRW;
Others: PEXTRW, PMULHU, PSHUFW, PMOVMSKB

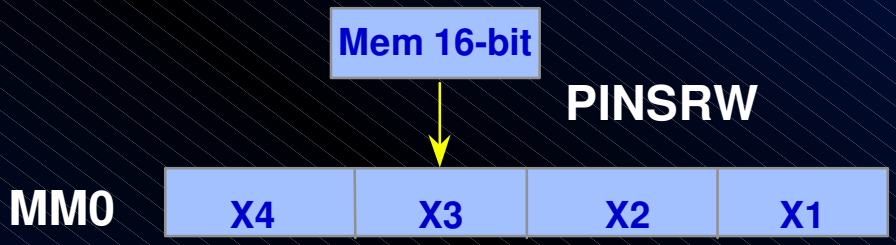
DVD Decode	6% (7%)
MPEG-2 Encode	Real Time @ 30fps (500/100 – 30%)
Speech Recognition	22% App. Gain

New Media Instructions

- Byte Mask Write - MASKMOVQ
- PMULH Unsigned - PMULHUW
- Insert/Extract - PINSRW/PEXTRW
- Move Mask - PMOVMSB
- Shuffle - PSHUFW
- Average - PAVGB/PAVGW
- Sum of Absolute Difference - PSADBW
- Minimum - PMINSW/PMINUB
- Maximum - PMAXSW/PMAXUB

PINSRW / PEXTRW

Immediate Specifies Which MMX™ Technology Operand to Insert into



Immediate Specifies Which MMX Technology Operand to Extract from

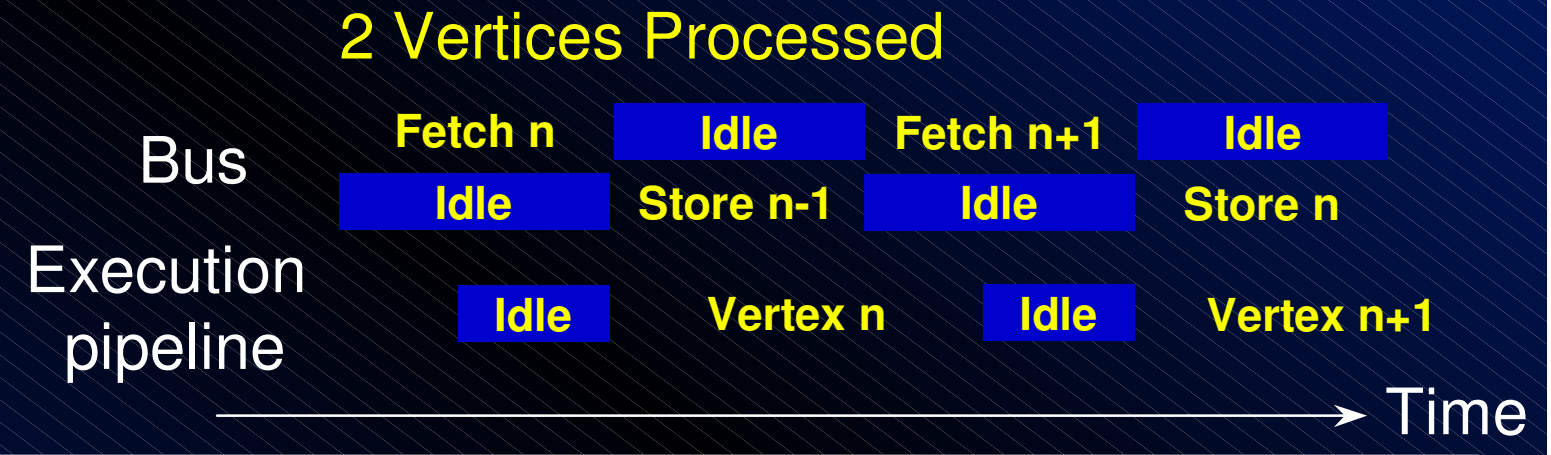
Memory Latency - Big Issue

- 3D scenes don’t fit in L2 cache
- Latency limits throughput
- Cache pollution hurts performance

ISSE Enables Concurrency

Load vertex data, transform & light, return to memory

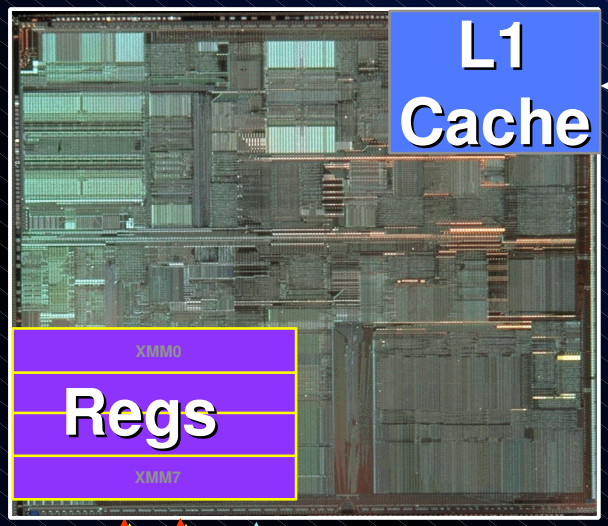
Without memory streaming or SIMD-FP



With memory streaming & SIMD-FP

Concurrency Hides Impact of Memory Latency

Pentium® III Platform



Prefetch/Streaming Store

- Prefetch - architectural state not updated
  - specify cache(s) to place prefetched data
- Streaming Store - non-allocating write
  - Provide WC memory type per store
  - Avoid cache pollution for write-once data
- SFENCE primitive to flush WC lines

Prefetching is Applicable to all OS, Apps.

New State Related

- New State Save & Restore
  - FXSAVE, FXRSTOR
- Load and Store Control Status Register
  - STMXCSR, LDMXCSR

Outline

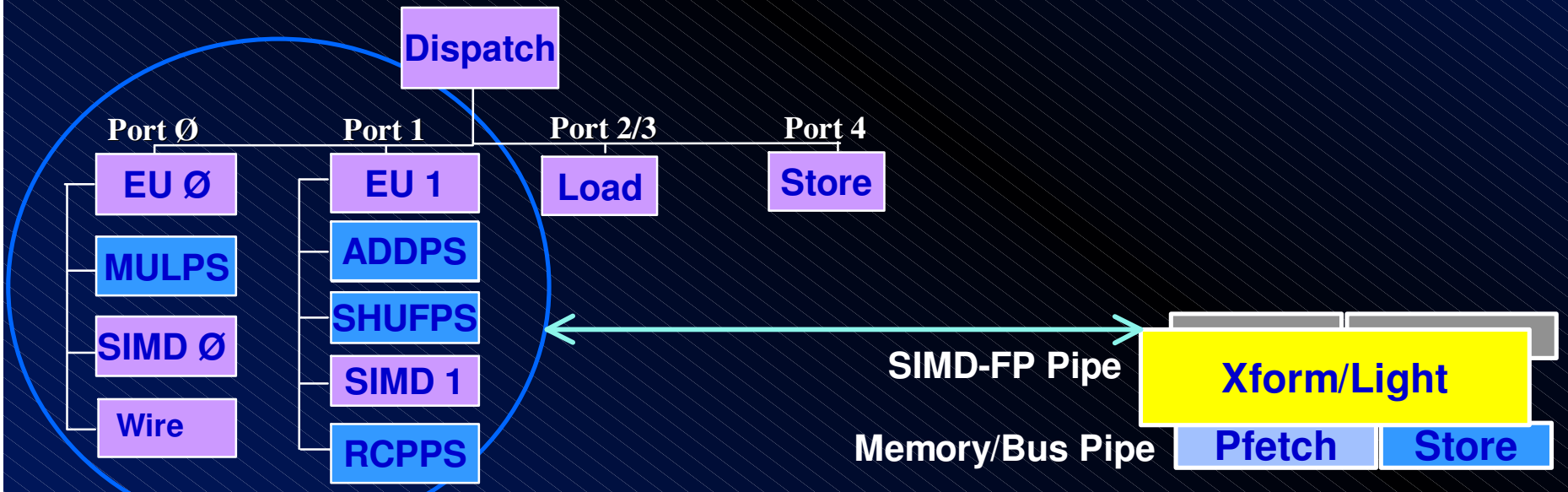
- Goals
- Architecture
- Instruction Set Arch./Benefits
- **Status**
- Conclusion

ISSE Implementation

- Definition completed in 4Q96
- ISSE implemented in 4 designs
  - IA-32: Pentium® III proc., Willamette (Perf.)
  - IA-64: Merced, McKinley (Emulation)
- Pentium III processor launched in 1Q99
- Die Cost:
  - Pentium III proc.(10%), Rest (<1.5%)

Pentium® III Proc. Implement.

- Packed Add and Multiply on Different Ports, Reciprocal
- Increased Shuffling/Data Movement and Throughput
- 20% higher bus write throughput (800MB/s)
- Increase Memory Throughput by adding More WC Store Buffers & Improving Allocation Policy



Improved Peak Throughput and Utilization of SIMD-FP

Conclusion

- Native SIMD-FP/Int support in processor accelerates human interface apps
  - ISV innovation's exceed expected gains
  - Performance scales with frequency
- Pentium® III proc. design balanced between perf, die cost & schedule
  - Provides app. gains beyond Moore's law
- 4-wide SIMD-FP architectures provides headroom for future implementations