EE282
Computer Architecture

Lecture 13:
Media Instruction Set Extensions

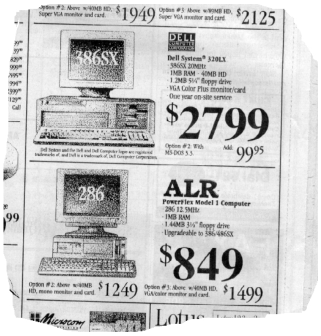
November 13, 2001

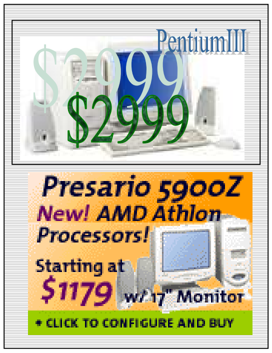
Parthasarathy (Partha) Ranganathan

Organization of class

- Motivation and media processing workloads
  - Media processing scenarios
  - Media processing workload characteristics
  - Media processing architecture continuum
- ISA Extensions for media processing
  - Common features of media ISA extensions
  - Differences between media ISA extensions
  - Intel MMX and SSE
  - Performance benefits from media ISA extensions
- Summary

Motivation - Moore’s Law


Option #2: Above w/40MB HD, Super VGA monitor and card. $1949
Option #3: Above w/89MB HD, Super VGA monitor and card. $2125
386SX
DELL COMPUTER CORPORATION
Dell System® 320LX
· 386SX 20MHz
· 1MB RAM · 40MB HD
· 1.2MB 5¼" floppy drive
· VGA Color Plus monitor/card
· One year on-site service
$2799
Option #2: With MS-DOS 3.3. Add: 99.95
Dell System and the Dell and Dell Computer logos are registered trademarks of, and Dell is a trademark of, Dell Computer Corporation.
286
ALR
PowerFlex Model 1 Computer
· 286 12.5MHz
· 1MB RAM
· 1.44MB 3½" floppy drive
· Upgradeable to 386/486SX
$849
Option #2: Above w/40MB HD, mono monitor and card. $1249
Option #3: Above w/40MB HD, VGA/color monitor and card. $1499
Microcom
Lotus


PentiumIII
$2999
$2999
Presario 5900Z
New! AMD Athlon Processors!
Starting at $1179 w/ 17" Monitor
▸ CLICK TO CONFIGURE AND BUY

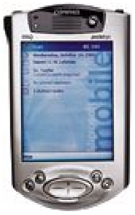
- Moore’s law – processor performance doubles every 18 months
  - 1990-2001: ~128X performance for same cost

Motivation

- Implications of Moore’s law: Every 18 months,
  - Performance doubles for same cost
  - Cost halves for same performance
- New applications and appliances enabled by these trends
  - Media processing – image, video, and audio
  - Graphics and user interfaces – games, speech recognition
  - Information processing – transaction processing, data mining
  - Web/internet applications

Sample Media Processing Scenarios (1)

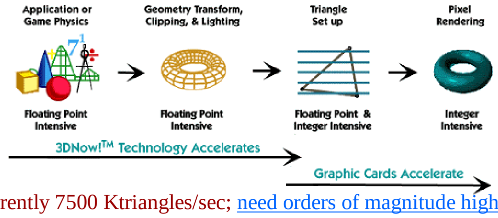
- Video conferencing, medicine, education, …
  - Currently, H.263 video over modem -- small images
  - HDTV : 720x485x3-band @30 frames/sec = 140Mb/s
  - Image compression -- very compute intensive



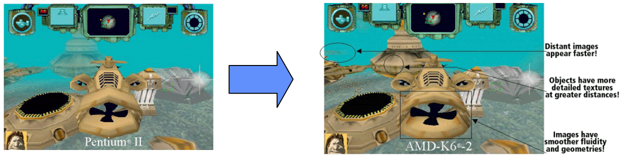
- Compaq iPAQ H3870
  - 206MHz StrongARM SA-1110+64MB SDRAM
  - PocketPC2002: Wireless+email+calendar+games
  - Next-generation: orders of magnitude more performance

Sample Media Processing Scenarios (2)

- Games and graphics-based applications
  - Geometry pipeline computation


Application or Game Physics
Geometry Transform, Clipping, & Lighting
Triangle Set up
Pixel Rendering
Floating Point Intensive
Floating Point Intensive
Floating Point & Integer Intensive
Integer Intensive
3DNow!TM Technology Accelerates
Graphic Cards Accelerate

  - Currently 7500 Ktriangles/sec; need orders of magnitude higher performance
- From the AMD 3DNow! Web pages


Pentium® II
AMD-K6®-2
Distant images appear faster!
Objects have more detailed textures at greater distances!
Images have smoother fluidity and geometries!

Sample Media Processing Scenarios (3)

- Immersive and interactive virtual environments
  - Immersadesk: immsersive, 3D visualization
  - 4 195MHz R10000 CPUs, 3.3 GB memory,
  - InfiniteReality2 graphics, IRIS audio Processors
  - Very highly compute-intensive
- 3D displays - emerging technology
  - auto-sterereoscopic displays (without glasses) need dual rendering

Multimedia Scenarios, Tasks, and Functions

- Media processing scenarios
  - Workgroup collaboration, video cataloging, digital library, tele-medicine, remote diagnosis, remote conferencing, speech-recognition with indexing, media data mining, real-time video authoring, wearable computing, internet robots, avatars, …
- Media processing tasks
  - Image, Video, Audio, Speech, Graphics, Communication
- Media processing functions
  - Matrix operations, signal processing functions (e.g. FFT), image processing functions (quantization, motion estimation), bit-level arithmetic, ...

Benchmarks

- To build architectures, we have to understand applications
  - SPEC -- Standard Performance Evaluation Corporation
    - http://www.spec.org
    - CPU SPECINT and CPU SPECFP
    - Mainly engineering and scientific workloads, not media-centric
  - Media workloads: Intel Media Bench, Norton Media Benchmark, BDTI, Ziff-Davis Graphics Benchmark
- The two classes of workloads are very different!

Media processing workload characteristics (1)

- Continuous media data types
  - Data elements derived from sampling analog input
  - Fundamentally different from those for general apps
    - correctness dictated by human perception
    - smaller media data types (8-bit, 16-bit)

      how does this compare to data paths in current processors? (32 or 64-bit)
  - Important to be able to do sub-word processing
- Real-time and peer-to-peer
  - More stringent computational thresholds
    - e.g., 30 frames per second for video

Media processing workload characteristics (2)

- Significant levels of fine-grained data parallelism
  - High levels of inherent data parallelism
    - large collection of small data elements
    - identical processing of similar elements
    - e.g. Image Addition -- *what does this imply for architecture?*

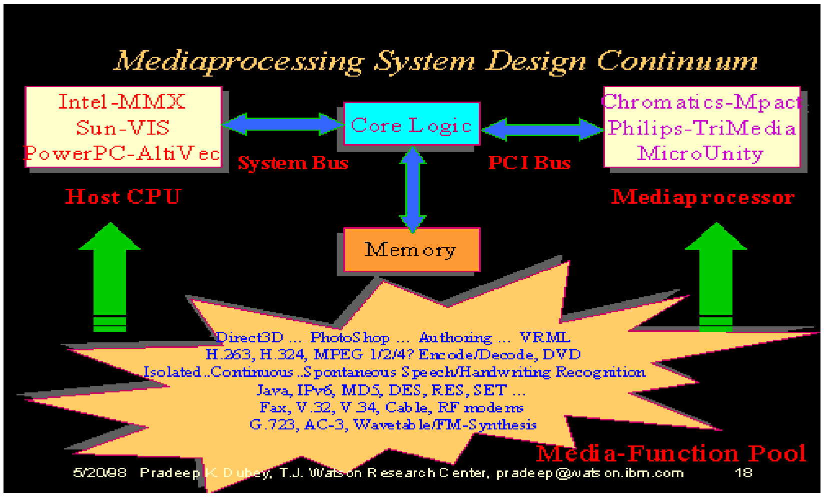
```
For I = 1 to 1024                    Dest1 = src11+src21
For J = 1 to 1024          ⇨         Dest2 = src12+src22
   dest[I,J]                         Dest3 = src13+src23
      = src1[I,J]+src2[I,J]          Dest4 = src14+src24
                                     ……
```

- Significant coarse-grain parallelism
  - High thread-level parallelism within application
  - More than one time-critical application at a time
    - e.g., wireless teleconferencing: video, audio, speech, wireless, graphics

Media processing workload characteristics (3)

- High memory bandwidth needs
  - Low data cache locality
  - e.g.: video: 3-band 720x485 @30frames/sec = 140Mbps
    audio: 60 minutes of two-channel 16-bit audio @44.1KHz = 605Mbps
- High network bandwidth needs
- Extensive data reorganization requirements
  - E.g., FFT butterfly algorithm
- Higher instruction-cache locality (small loops)

Media processing System Design Continuum


Mediaprocessing System Design Continuum
Intel-MMX
Sun-VIS
PowerPC-AltiVec
Host CPU
System Bus
Core Logic
PCI Bus
Chromatics-Mpact
Philips-TriMedia
MicroUnity
Mediaprocessor
Memory
Direct3D ... PhotoShop ... Authoring ... VRML
H.263, H.324, MPEG 1/2/4? Encode/Decode, DVD
Isolated..Continuous..Spontaneous Speech/Handwriting Recognition
Java, IPv6, MD5, DES, RES, SET ...
Fax, V.32, V.34, Cable, RF modems
G.723, AC-3, Wavetable/FM-Synthesis
Media-Function Pool
5/20/98 Pradeep K Dubey, T.J. Watson Research Center, pradeep@watson.ibm.com 18

Architectures for Media Workloads

- Several alternatives
  - Special-purpose processors
  - Digital signal processors
  - Multimedia co-processor
  - Media processor
  - General-purpose processors with media extensions
  - Media processors with general-purpose extensions
- Today, we will focus only on GPPs with media extensions
  - Current trends: *programmable, general-purpose*

Media ISA Extensions to General-Purpose Processors

- Key idea: add instruction support to speed up media tasks

  First used in Intel i860 (‘89) -- 6 instructions for 3D triangle rendering and the HP PA7100LC (circa 90) -- 5 instructions using implementation-specific features

  – Integer/Fixed-point extensions
    - HP MAX (Media Acceleration eXtensions), Sun VIS (Visual Instruction Set), Intel/Cyrix/AMD MMX (multi-media extensions), MIPS MDMX (MIPS Digital Media Extensions), Digital MVI (Motion Video Extensions), (IBM/Motorola) PowerPC AltiVec technology

  – Floating-point extensions
    - AMD 3DNow!, Intel SSE (internet and streaming extensions), MIPS V extensions, PowerPC AltiVec

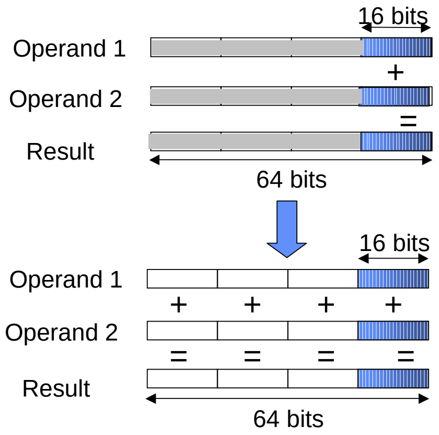
- Constraints: ISA changes affect all levels of architecture
  – Need to have significant performance benefits for target applications
  – No adverse performance impact on other applications
  – Scaleable benefits and implementation complexity in future

Common Features of Media ISA Extensions

- Extensions for packed data types
- SIMD (vector-style) instructions
- Support for saturation arithmetic
- Extensions for sub-word rearrangement and alignment
- Extensions for conditional execution
- Extensions for memory bandwidth optimizations
- Special-purpose operations

Packed data types and SIMD instructions

Motivation: wasted bits in current data paths



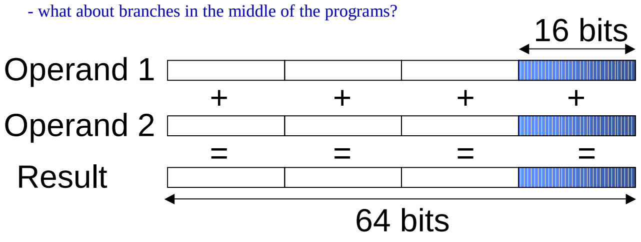
48 bits are wasted!

Can we use them in any way?

4 operations in 1 cycle
SPEEDUP: 4X

Packed Data Types and SIMD Parallelism

- – This is called **SIMD** (single-instruction-multiple-data) parallelism
- – The data types are called *packed data types*
- – *What are the tradeoffs?*
  - + exploits small data types in media workloads
  - + exploits data parallelism in media workloads (remember image addition example)
  - + small changes to implementation (almost for free!)
  - - packing and unpacking data
  - - how do we handle overflow?
  - - what about branches in the middle of the programs?

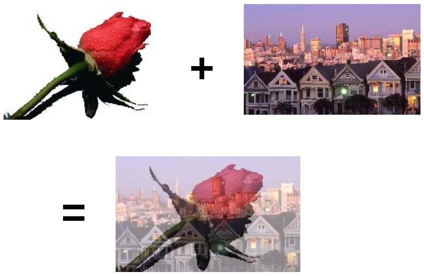


Extensions for Saturation Arithmetic

- Why do we need saturation arithmetic?
  - Many media applications spend a large fraction of time checking boundary conditions to ensure clamping of values to maximum of minimum.
  - Example: image addition

```
For I = 1 to 1024
For J = 1 to 1024
   dest[I,J]
      = src1[I,J]+src2[I,J]

   If (dest > 255)
     dest = 255;
   If (dest < 0)
     dest = 0;
```



  - Saturation ensures clamping of values;
    - Support saturation arithmetic in ISA directly
    - Saves a number of instructions

Extensions for Sub-word Rearrangement

- How do we go from unpacked data to packed data types?
  - Provide ISA support for pack, unpack, expand, align, …
  - Remember that these add to the overhead instructions, so make sure they occur infrequently; for e.g., once at the beginning of the loop

- Support for other types of sub-word rearrangement
  - Shift, rotate, permute, mix-and-permute, ...
  - *Why would we need them?*
  - Many algorithms need these -- e.g., FFT butterfly algorithm

Extensions for Conditional Execution

- Why do we need support for conditional execution?
  - Many applications need this
  - Consider chroma-keying (weather person on radar screen)

```
If (src1 is green)
 dest = src2;
else
  dest = src1;
```

  - *How do we implement this with SIMD?*
  - `Cmp,simd src1, green, mask1`
  - `and,simd mask, src1, dest, dest`
  - `and,simd !mask, src2, dest, dest`
  - Notice how the control dependence has been converted to data dependence and we still can extract SIMD parallelism.

Other Extensions

- Memory-related extensions
  - Recall memory bandwidth requirements of media workloads
  - Use prefetching, provide cache hints on allocation, replacement
  - Sun VIS provides special instructions for blocked loads and stores
    - *What are the tradeoffs?*
- Special-purpose Instructions
  - Targeted at specific compute-intensive parts of media workloads
  - E.g., support for motion estimation in video encoding (VIS, MVI, SSE)
    - Perform sum-of-absolute-differences between image frames
    - *What are the tradeoffs of adding such special-purpose instructions?*

Differences between Media ISA Extensions

- Why are there differences?
  - Additions to existing ISAs are constrained by number of bits available, number of existing registers, current data path, and existing support for instructions
- Evolution versus revolution
  - Typically marginal impact on chip area and clock frequency. E.g., MAX (0.2%), VIS (3%), MMX (1%), MVI (0.6%), SSE (10%). Altivec is exception; takes more area.
- Number of Instructions supported
  - E.g., Altivec: 162 instructions, SSE: 70 instructions, MVI: 13 instructions
- Fixed-point versus floating-point instructions
  - Int: MAX, VIS, MMX (multi-media extensions), MDMX, MVI
  - FP: 3DNow!, MIPSV
  - Both: AltiVec

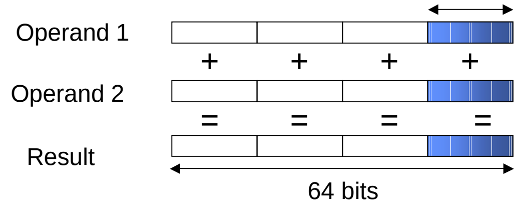
Differences between Media ISA Extensions

- Implementation data path and state for fixed-point
  - Integer versus floating-point data path
    - MAX and MVI use integer register file and data path; MMX, VIS, and MDMX use floating-point data path
    - Altivec and SSE add a new data path
      - Integer data path, FP data path, media data path
  - Tradeoffs:
    - Performing video/audio (integer) and graphics (FP) operations in parallel
    - Performing addressing/branching (integer) and video/audio (FP) in parallel
    - Separate data paths allow greater data widths.
    - How about complexity and scalability of implementation?
  - Additional hardware state
    - e.g., graphics status register (VIS), vtselect field and larger accumulator with MDMX, control status registers with AltiVec -- *what are the tradeoffs?*

Differences between Media ISA Extensions

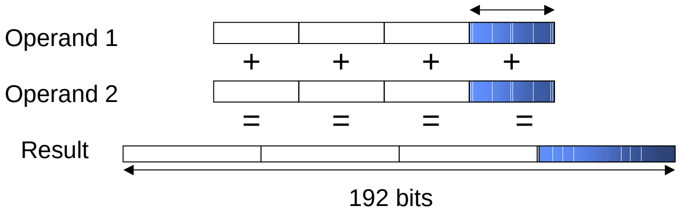
- Packed data types and SIMD parallelism
  - Depends on data path size (32-bit versus 64-bit)
  - Depends on accuracy required

    how does this relate to media types?
  - Impact parallelism extractable
  - E.g., MMX, VIS, AltiVec support 8-, 16-, and 32-bit data types, MDMX and MVI support 8- and 16-bit packed data types, MIPS V, 3DNow!, AltiVec, SSE support packed single data types

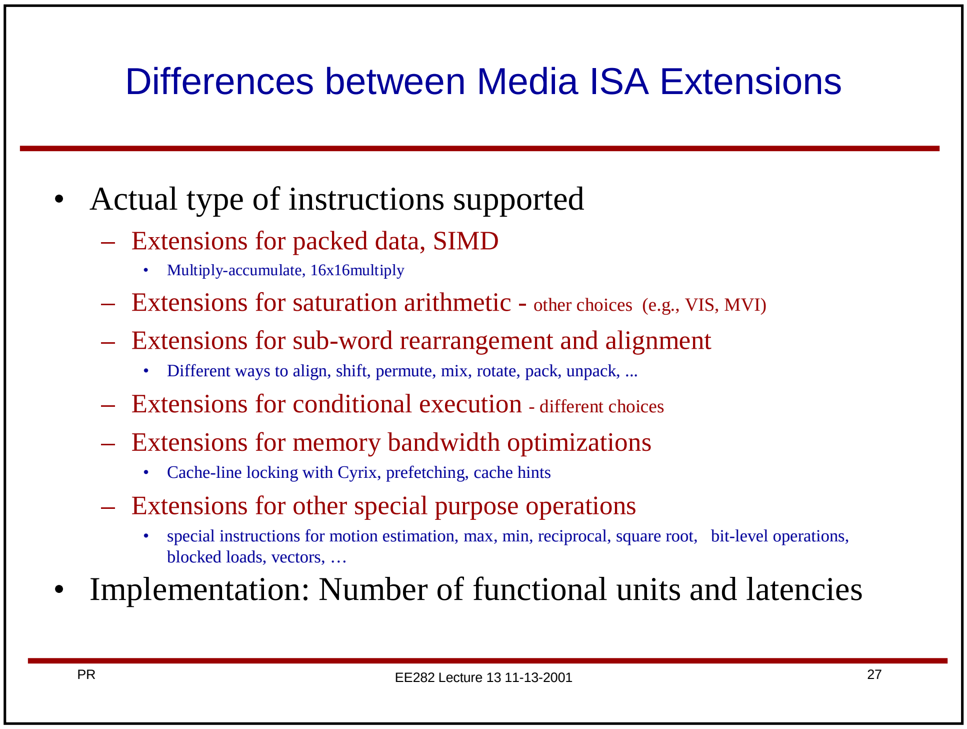


Differences between Media ISA Extensions

- MIPS V trick -- larger accumulator

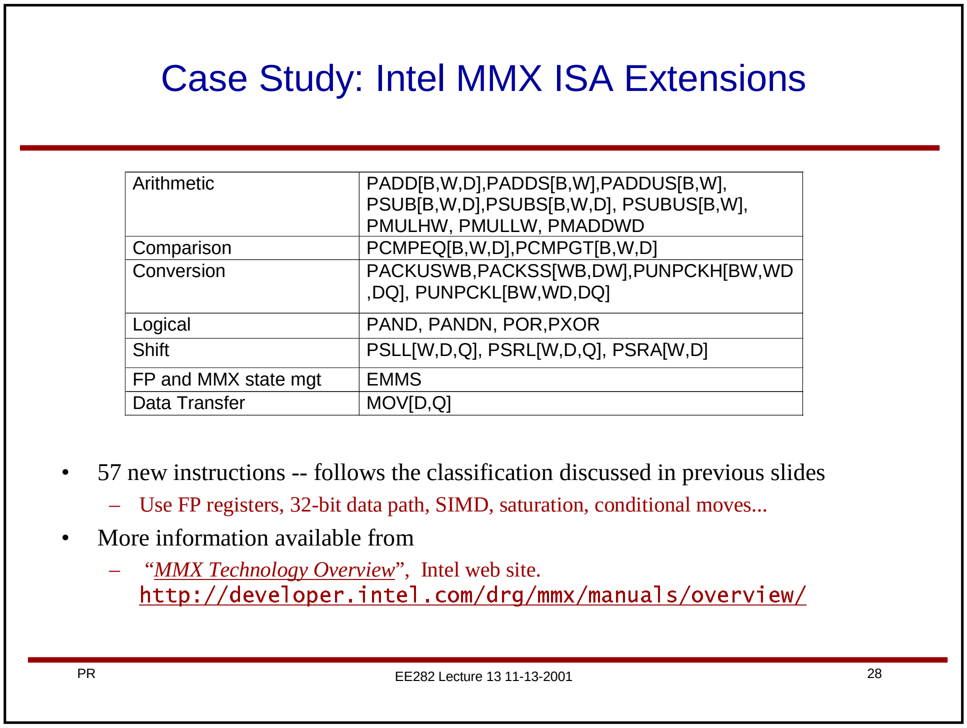


- *What are the tradeoffs?*
  - Better accuracy AND more parallelism
  - More complicated implementation
  - Only accumulator is 192-bit long, possible register pressure for this register



Differences between Media ISA Extensions

- Actual type of instructions supported
  - Extensions for packed data, SIMD
    - Multiply-accumulate, 16x16multiply
  - Extensions for saturation arithmetic - other choices (e.g., VIS, MVI)
  - Extensions for sub-word rearrangement and alignment
    - Different ways to align, shift, permute, mix, rotate, pack, unpack, ...
  - Extensions for conditional execution - different choices
  - Extensions for memory bandwidth optimizations
    - Cache-line locking with Cyrix, prefetching, cache hints
  - Extensions for other special purpose operations
    - special instructions for motion estimation, max, min, reciprocal, square root, bit-level operations, blocked loads, vectors, …
- Implementation: Number of functional units and latencies



Case Study: Intel MMX ISA Extensions

Arithmetic	PADD[B,W,D],PADDS[B,W],PADDUS[B,W], PSUB[B,W,D],PSUBS[B,W,D], PSUBUS[B,W], PMULHW, PMULLW, PMADDWD
Comparison	PCMPEQ[B,W,D],PCMPGT[B,W,D]
Conversion	PACKUSWB,PACKSS[WB,DW],PUNPCKH[BW,WD,DQ], PUNPCKL[BW,WD,DQ]
Logical	PAND, PANDN, POR,PXOR
Shift	PSLL[W,D,Q], PSRL[W,D,Q], PSRA[W,D]
FP and MMX state mgt	EMMS
Data Transfer	MOV[D,Q]

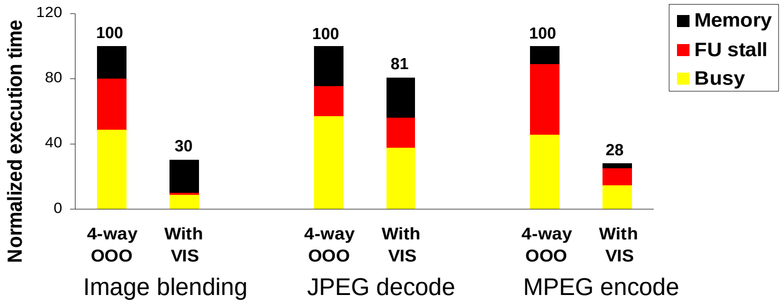
- 57 new instructions -- follows the classification discussed in previous slides
  - Use FP registers, 32-bit data path, SIMD, saturation, conditional moves...
- More information available from
  - "*MMX Technology Overview*", Intel web site. http://developer.intel.com/drg/mmx/manuals/overview/

Case Study: Intel SSE ISA Extensions

Data movement	MOV, MOVUPS, MOVLPS, MOVLHPS, MOVHPS, MOVHLPS, MOVMSKPS, MOVSS
Shuffle	SHUFPS, UNPCKHPS, UNPCKLPS
State	FXSAVE, FXRSTOR, STMXCSR, LDMXCSR
MMX Tech Enhancements	PINSRW, PEXTRW, PMULXHU, PSHUFW, PMOVMSKRB, PSAD, PAVG, PMIN, PMAX
Streaming/prefetching	MASKMOVQ, MOVNTQ, MOVTPS, PREFETCH, SFENCE
Conversions	CVTSS2SI, CVTTSS2SI, CVTSI2SS, CVTPI2PS, CVTPS2PI, CVTTPS2PI

- 70 instructions -- follows the format discussed in previous slides
  - Separate register state, 128-bit data path, alignment support, cache hints, SIMD,...
- More information available from
  - "*The Internet Streaming SIMD Extensions*", Shreekanth Thakkar and Tom Huff, Intel Technology Journal Q2, 1999. http://developer.intel.com/technology/itj/q21999/articles/art_1.htm

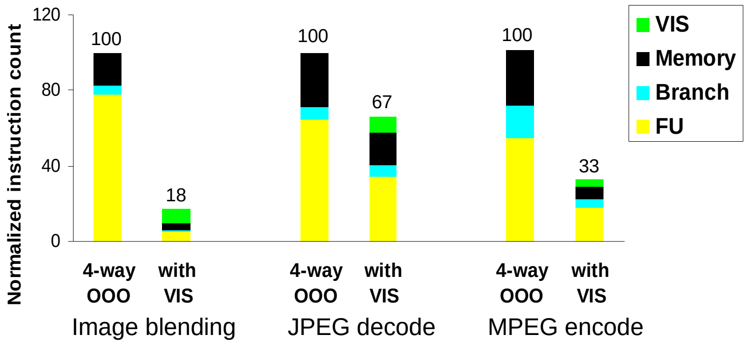
Impact of Media ISA Extensions [ISCA99]



The results presented in the following slides are from Ranganathan,Adve,Jouppi, ISCA99. The results compare a state-of-the-art 4-way issue system with dynamic scheduling (out-of-order execution) without and with the SPARC VIS media ISA extensions. 12 image and video benchmarks were studied using the RSIM simulator; only three are presented here.

- Media ISA extensions improve performance (1.1X to 4.2X)

Instruction Count Reduction with Media ISA Extensions



- Instruction counts correlate well with performance improvements
- Reductions in FU, branch, memory instructions
  - Main reductions from FU instructions

Instruction Count Reduction with Media ISA Extensions

- FU instructions (SIMD)
- Branch instructions (Saturation, Edge, SIMD)
  - Reduced branch misprediction rates
- Memory instructions (SIMD)
  - Reduced contention
  - Increased cache miss rates
- Special instruction for mpeg encode (pixel error)
  - 48 instructions replaced by one instruction
  - Reduced branch misprediction rate (27% to 10%)

Limitations of Media ISA Extensions

- Inapplicable in some cases (*e.g.*, Huffman coding)
- Too specialized (*e.g.*, pixel error)
- Too much overhead (packing and alignment)
- Limited by 64-bit data path (*e.g.*, over 16-bit accuracy)
- Do not directly target cache miss latency
  - In many cases, memory starts to become a problem

(Note: some of these problems are specific to the media extension considered in this particular study.)

Review

- Media processing emerging to be important workload
- Variety of architectural choices emerging
  - Today, media ISA extensions for general-purpose processors
  - More links related to ISA extensions can be found on web
- As media workloads become ubiquitous, more aggressive computer architectures required
  - Interesting work going on here at Stanford

Next Class

- More on caches
- Memory systems