Native Signal Processing

Ravi Bhargava
Laboratory of Computer Architecture
Electrical and Computer Engineering Department
The University of Texas at Austin

November 22, 1999

Motivation

- Need new ways to support emerging applications
  - Multimedia
  - Signal processing
  - 3D graphics
- Many applications have similar properties
  - Large streams of data
  - Data parallelism
  - Small data types
  - Tight loops
  - High percentage of computations
  - Common computations (e.g. MAC)

Target Applications

- IP telephony gateways
- Multi-channel modems
- Speech processing systems
- Echo cancelers
- Image and video processing systems
- Scientific array processing systems
- Internet routers
- Virtual private network servers

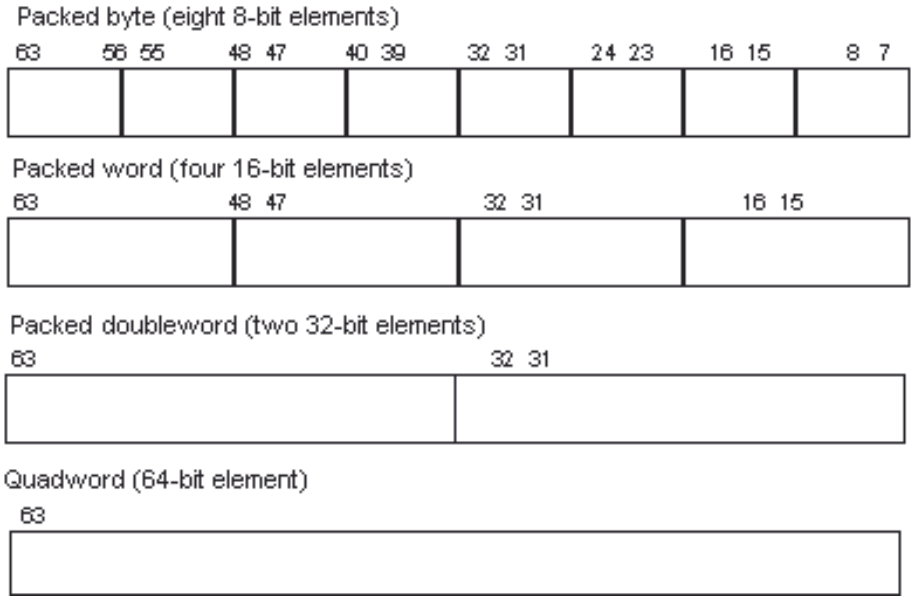
Basic Idea

- Extend capabilities of general-purpose processor
  - Additions to the instruction set architecture
  - Additions to the CPU state

- Exploit Data Parallelism
  - Single Instruction Multiple Data (SIMD)
  - Multiple operations using one functional unit

SIMD

- Use one register to store many pieces of data



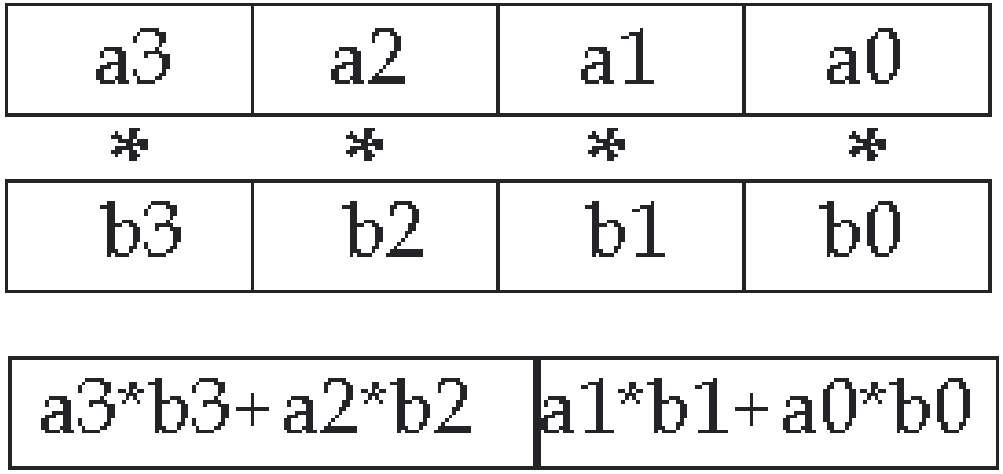
Packed Arithmetic

- Unsigned Packed Add with Saturation

a3	a2	a1	FFFFh
+	+	+	+
b3	b2	b1	8000h
a3+b3	a2+b2	a1+b1	FFFFh

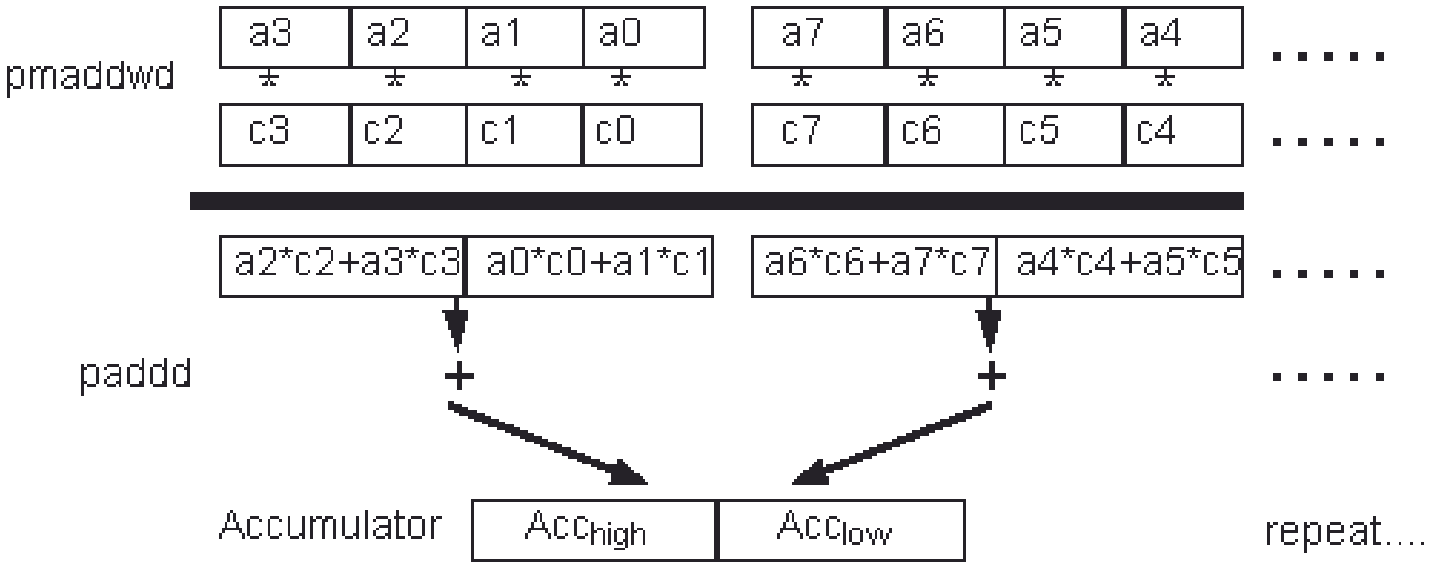
Packed Arithmetic

- Multiply Accumulate

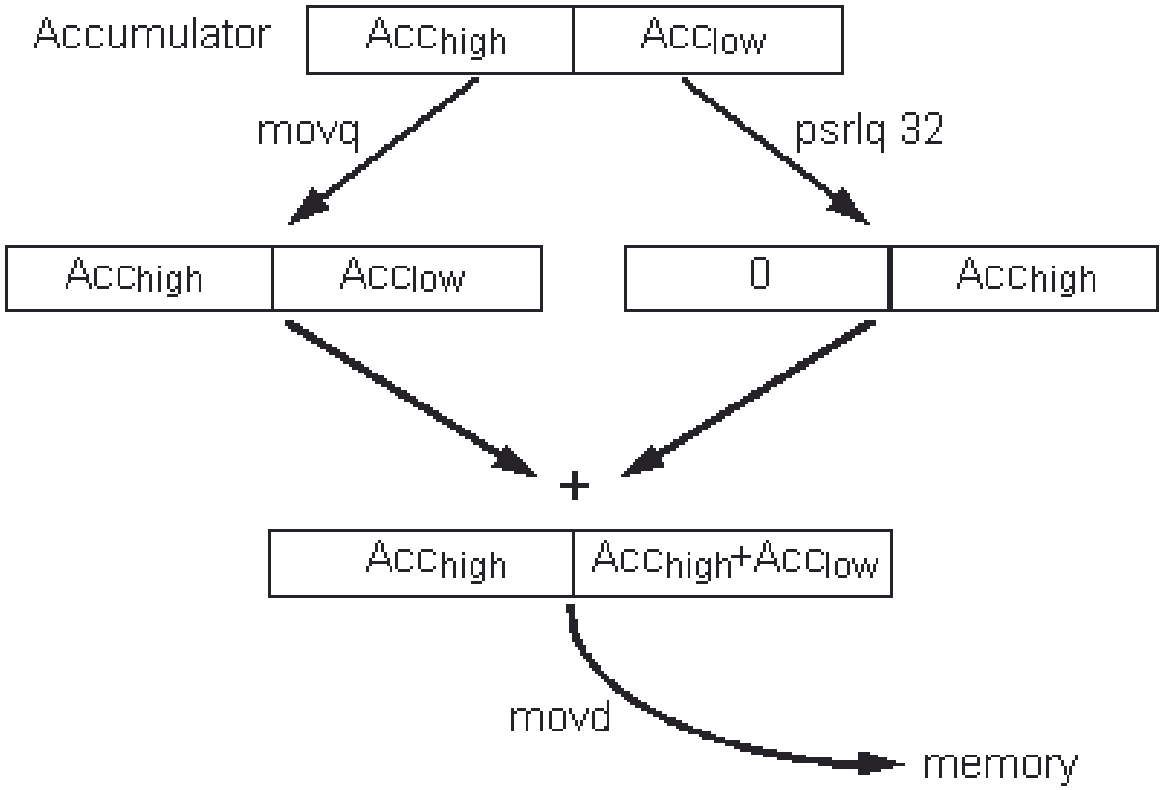


MMX Example

- 16-bit Vector Dot-Product

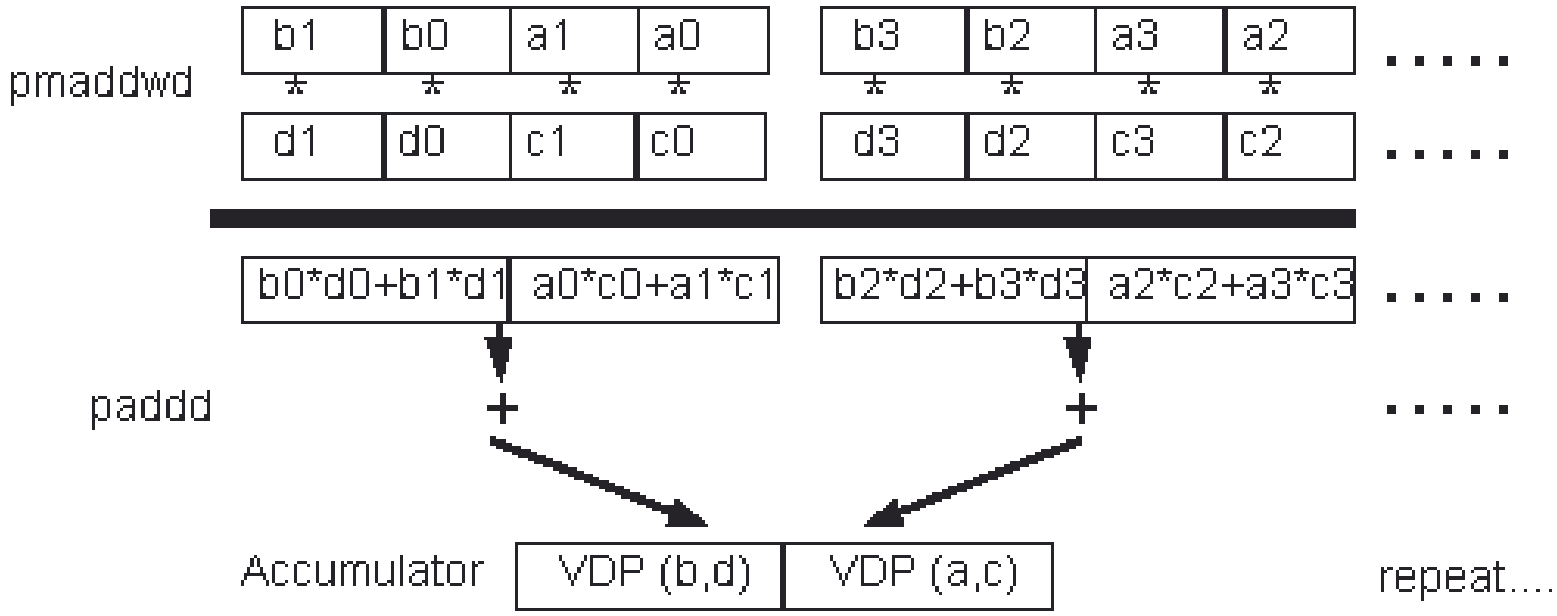


Dot-Product (cont.)



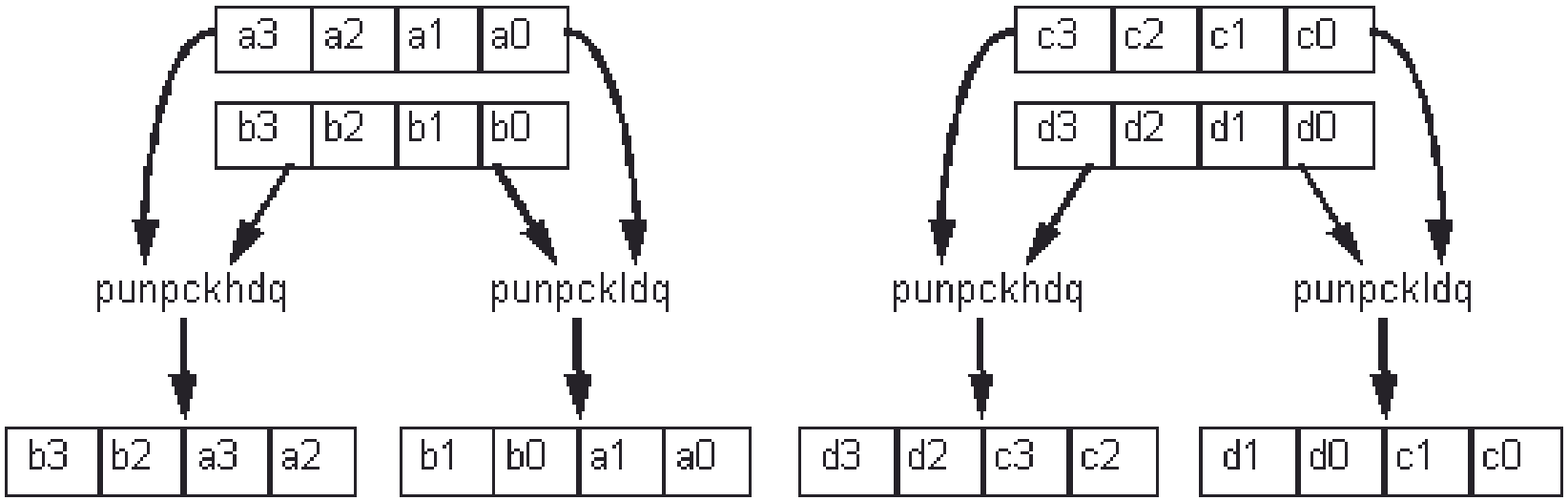
Alternative Dot-Product

- Useful in certain cases



Alternative Dot-Product (cont.)

- Interleaving the Data



Support for NSP

- Libraries
  - Most common type of support
  - Tradeoff speed for robustness
- Compiler
  - Can’t do too much, but getting better
  - Can use macros, annotations, and “intrinsics”
- Operating System
  - Needs to know about any new registers and states
  - Needs to know about any new exceptions

Common Concerns

- Organizing SIMD data
  - Packing and unpacking SIMD registers
  - Shuffling data within the register
  - “swizzling”, “scatter and gather”
  - Permute units
- Aligning data
  - Special load and store instructions
  - New data types in compilers
  - Padding data types

New Instructions

- Packed Comparisons
- Packed maximum, minimum
- Packed Average
- Packed Sum of absolute differences
- Multiply-accumulate
- Conversion: signed, data length, data type
- Approximations: reciprocal, reciprocal square root
- Prefetch
- Store fence

Evolution of NSP

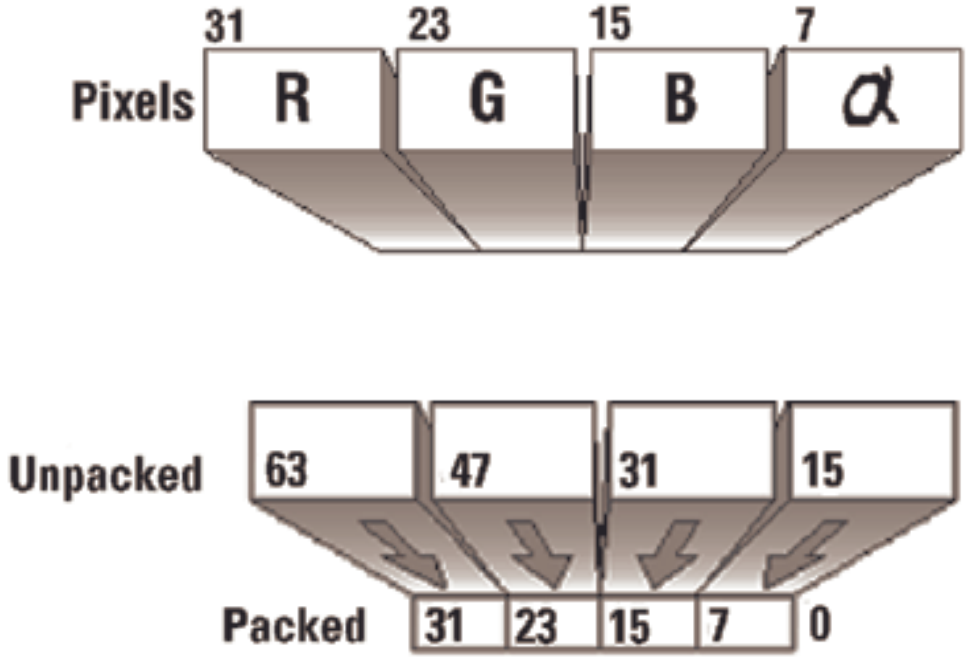
- Visual Instruction Set (VIS)
  - First NSP extension found on Sun's UltraSPARC
- MMX
  - First x86 NSP extensions, created for Intel's Pentium
- 3DNow!
  - Addition of FP SIMD instructions on AMD's K6-2
- Altivec
  - Adds new CPU State for Motorola's G4 PowerPC
- Streaming SIMD Extensions
  - New x86 FP SIMD for Intel's Pentium III

VIS

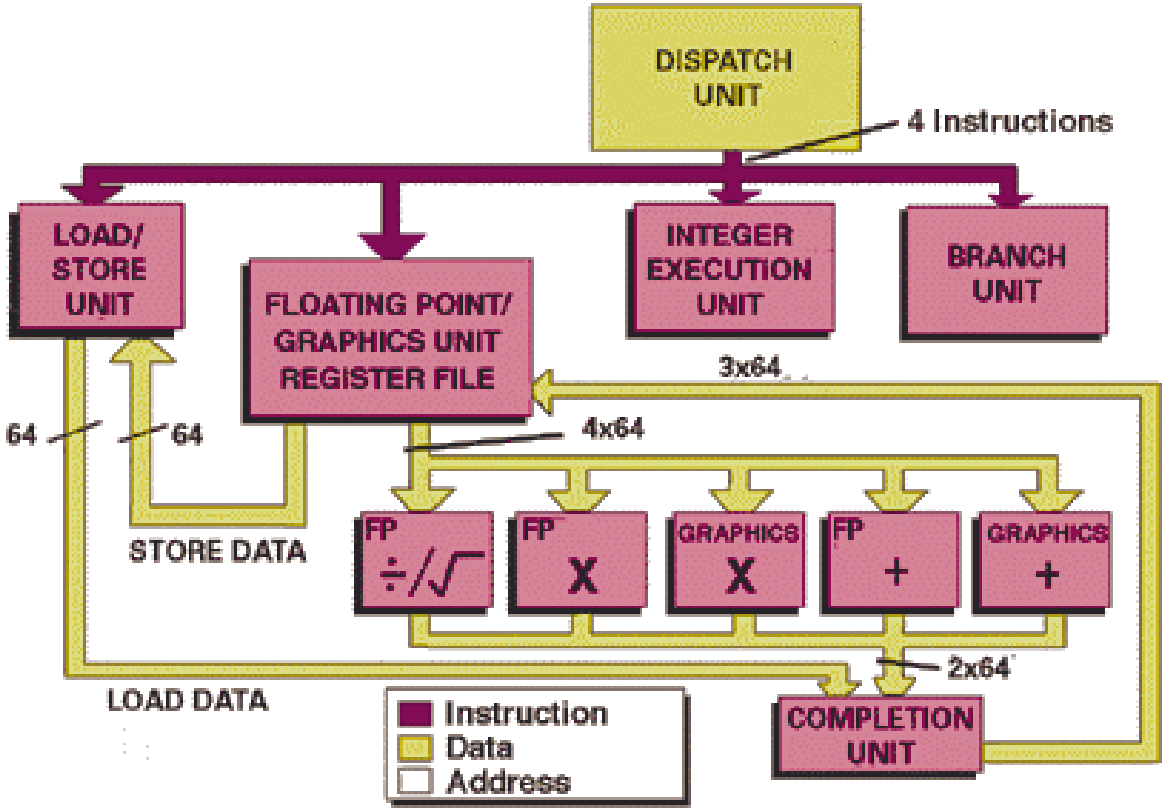
- Uses 64-bit floating point registers
- Data formats “tailored for graphics”
- No saturation arithmetic
- Only 8-bit x 16-bit multiplication
- Implicit assumptions about rounding & number of significant bits
- 3% increase in die area
- C Macros and “MediaLib” for development

VIS Tailored for Graphics

- Assumes pixels will be most common data



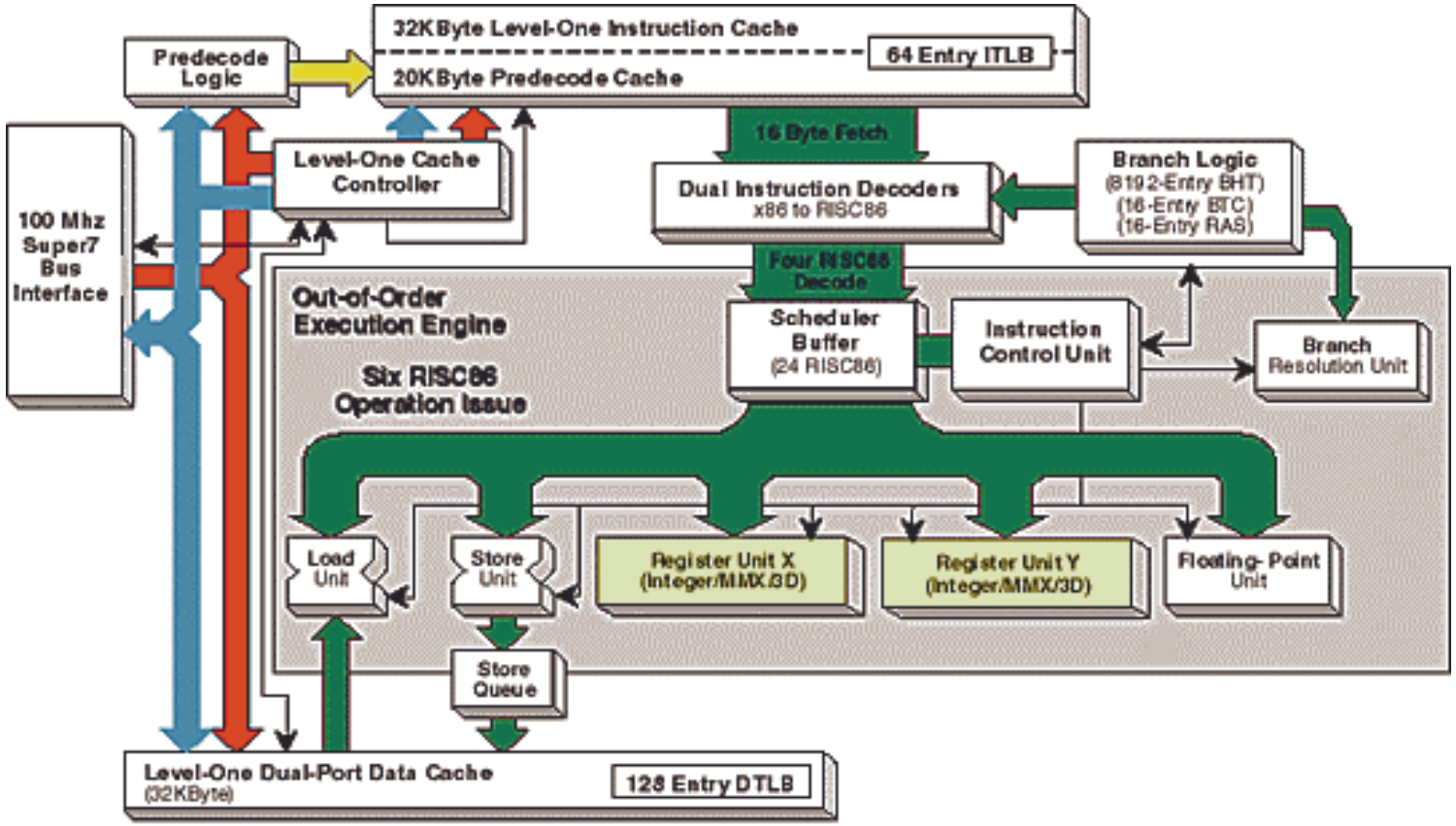
UltraSPARC with VIS



3DNow! on K6-II

- 21 new SIMD instructions
  - Including floating point SIMD instructions
  - PREFETCH instruction
  - Approximation and intra-element instructions
- Two 32-bit FP instructions in one register
  - Two 3DNow! instructions possible per cycle
  - Latency of two cycles, throughput of one cycle
- Aliased to the FP registers (like MMX)

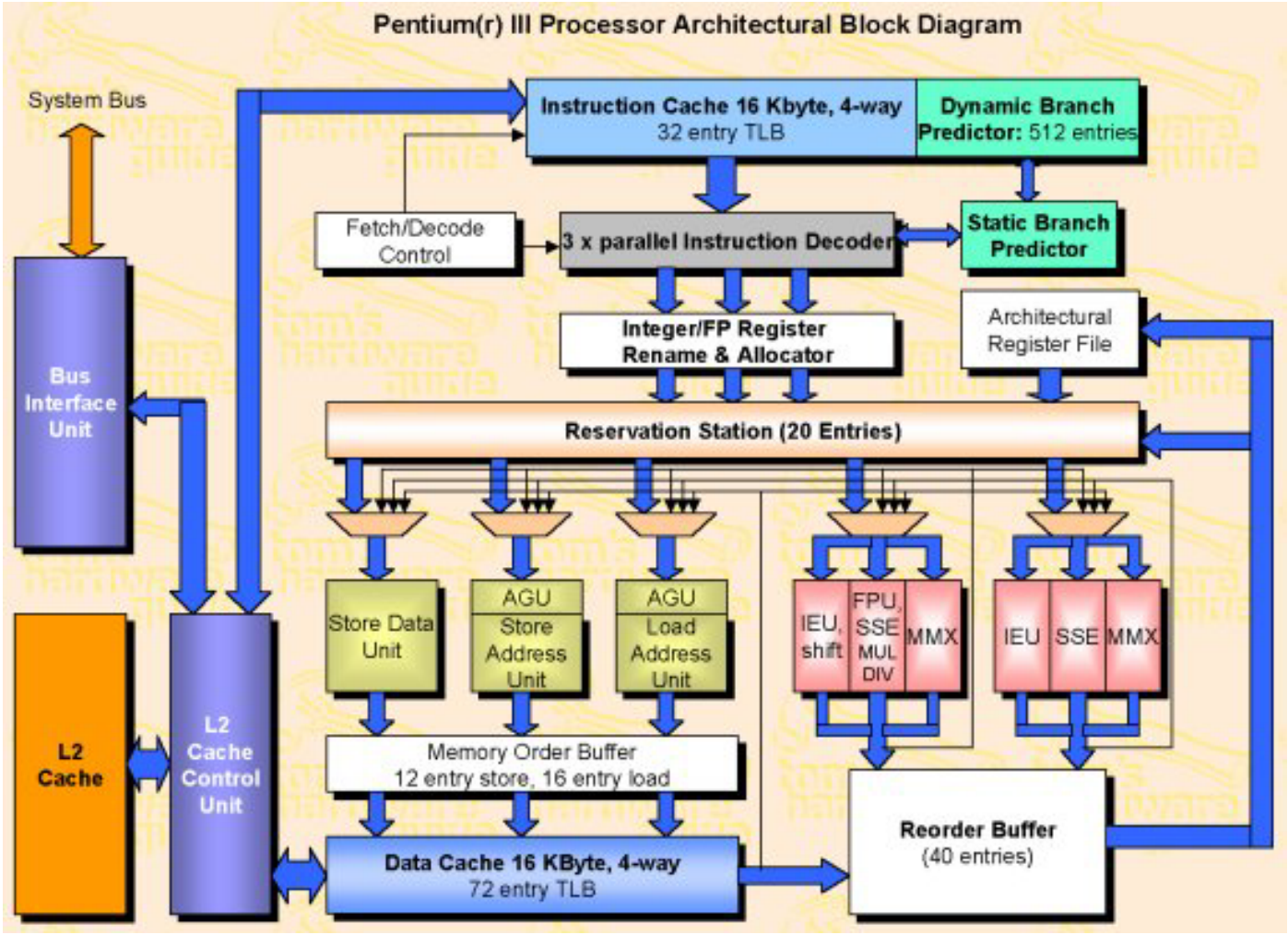
3DNow! on the K6-II



Streaming SIMD Extension

- 70 new instructions
- 8 new 128-bit SSE registers (OS support)
- Execute two SSE instructions simultaneously
  - Double-cycle through existing 64-bit hardware
  - 2 micro-ops per SSE instruction
  - Latency often two cycles or greater
- Explicit scalar SSE instructions
- Prefetch to various levels of cache
- Non-allocating stores
- 10% increase in die area

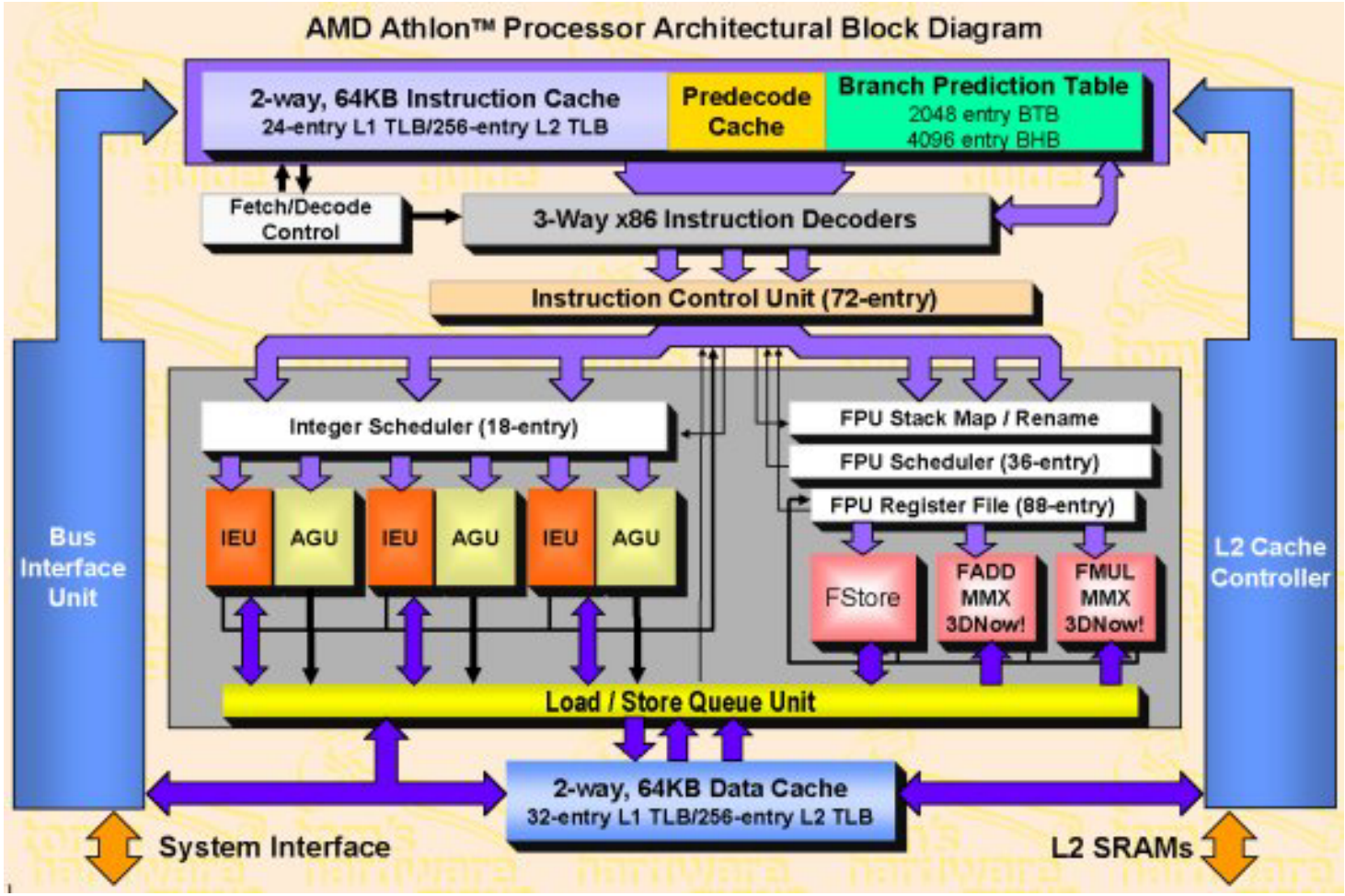
Pentium III


Pentium(r) III Processor Architectural Block Diagram
System Bus
Instruction Cache 16 Kbyte, 4-way
32 entry TLB
Dynamic Branch
Predictor: 512 entries
Fetch/Decode
Control
3 x parallel Instruction Decoder
Static Branch
Predictor
Integer/FP Register
Rename & Allocator
Architectural
Register File
Bus
Interface
Unit
Reservation Station (20 Entries)
Store Data
Unit
AGU
Store
Address
Unit
AGU
Load
Address
Unit
IEU,
shift
FPU,
SSE
MUL
DIV
MMX
IEU
SSE
MMX
L2
Cache
L2
Cache
Control
Unit
Memory Order Buffer
12 entry store, 16 entry load
Reorder Buffer
(40 entries)
Data Cache 16 KByte, 4-way
72 entry TLB

3DNow! on the Athlon

- Zero switching overhead back to FP!
- Most instructions have one cycle latency
- 24 more instructions
  - 12 more integer SIMD instructions
  - 7 more data manipulation instructions
  - 5 “unique” DSP instructions
    - Packed FP to Integer word conversion with sign extend
    - Packed FP negative accumulate
    - Packed FP mixed negative-positive accumulate
    - Packed integer word to FP conversion
    - Packed swap double-word

Athlon


AMD Athlon™ Processor Architectural Block Diagram
2-way, 64KB Instruction Cache
24-entry L1 TLB/256-entry L2 TLB
Predecode Cache
Branch Prediction Table
2048 entry BTB
4096 entry BHB
Fetch/Decode Control
3-Way x86 Instruction Decoders
Instruction Control Unit (72-entry)
Integer Scheduler (18-entry)
IEU
AGU
IEU
AGU
IEU
AGU
FPU Stack Map / Rename
FPU Scheduler (36-entry)
FPU Register File (88-entry)
FStore
FADD MMX 3DNow!
FMUL MMX 3DNow!
Load / Store Queue Unit
Bus Interface Unit
L2 Cache Controller
2-way, 64KB Data Cache
32-entry L1 TLB/256-entry L2 TLB
System Interface
L2 SRAMs

Altivec

- 162 new instructions
  - Intra-element: rotate, shift, max, min, sum across
  - Approximations: log2, 2 raised to an exponent
  - No explicit SAD instruction
- Separate 32-entry, 128-bit vector register file
- Floating-point and fixed-point data types
- Instructions have three source operands

Altivec (cont.)

- Permute Unit
  - Shuffle data from any of the 32 byte locations in two operands to any of 16 byte locations in the destination
  - Permute instructions - one cycle
- Vector Unit
  - Three sub-units
  - Simple fixed-point - one cycle
  - Complex fixed-point - three cycles
  - Floating-point - four cycles

Applications using 3DNow!

- 3D games and engines
  - Descent 4, Diablo 2, Quake III, Unreal 2
- 3D video card drivers
  - Matrox, Diamond, ATI, nVidia, 3DLabs, 3dfx
- Graphics APIs
  - OpenGL, DirectX, Glide
- Other apps
  - Naturally Speaking, LiveArt98, WinAMP, PicVideo, PC-Doctor, QuickTech, XingMP3

Additional Information

- MMX talk and additional references
  - http://www.ece.utexas.edu/~ravib/mmxdsp/
- NSP talk outline and links
  - http://www.ece.utexas.edu/~ravib/nsp/