Instruction Set extensions to X86

- Some extensions to x86 instruction set intended to accelerate 3D graphics
- AMD *3D-Now!* Instructions simply accelerate floating point arithmetic.
  - Accelerate object transformations
  - Allow multiple floating point operations to be done in one clock cycle.
- A similar extension is found on the Pentium III – just does not have the fancy name.

Floating Point SIMD instructions

- SIMD stands for Single Instruction, Multiple Data
- Same instruction applied to multiple operands
  - Do an add on four pairs of operands
    y0= a0 +b0, y1 = a1+b1, y2=a2+b2, y3 = a3+b3
- Pentium III added some 128 bit registers used to hold 'packed' single precision floating point numbers
  - A single precision floating point number is 32 bits

xmm Registers

New 128 bit registers are called XMM registers (XMM0 – XMM7)
Holds four 32-bit single precision floating point numbers

An instruction like ADDPS xmm0, ymm1 will add the two registers together, computing the sums of the four numbers.

Easy to see speed advantage over previous instructions

	4.0 (32 bits)	4.0 (32 bits)	3.5 (32 bits)	-2.0 (32 bits)
+	-1.5 (32 bits)	2.0 (32 bits)	1.7 (32 bits)	2.3 (32 bits)
	2.5 (32 bits)	6.0 (32 bits)	5.2 (32 bits)	0.3 (32 bits)

SIMD Extensions

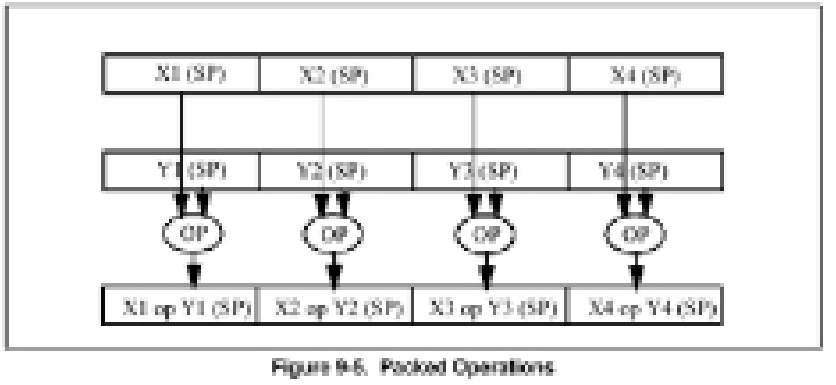


Figure 9-6. Packed Operations

More than 70 instructions. Arithmetic Operations supported: Addition, Subtraction, Mult, Division, Square Root, Maximum, Minimum. Can operate on Floating point or Integer data.

Flags

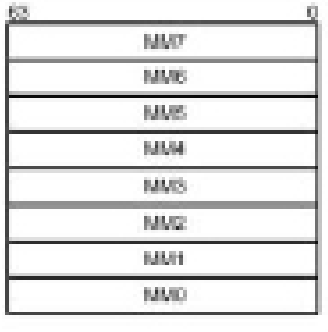
- Individual flags are not kept for each packed operation.
- Can only tell if an error (exception) occurred in one or more of the packed operations
- Some possible exceptions (not all listed)
  - Underflow (number too small)
  - Overflow (number too large)
  - Divide by Zero

Pentium 3 vs. Pentium 4

- The SIMD extensions on the Pentium 3 are called the SSE instructions and the 128 bit registers only support viewing the data as 4 single precision FP numbers.
- On the Pentium 4, the 128 bit registers can be viewed as these data types
  - 4 single precision FP values (SSE)
  - 2 double precision FP values (SSE2)
  - 16 byte values (SSE2)
  - 8 word values (SSE2)
  - 4 double word values (SSE2)
  - 1 128-bit integer value (SSE2)

MMX Instructions

Added eight 64 bit registers. The 64 bit register can be viewed as containing 8 packed bytes, 4 packed words, 2 dwords, or 1 quad.



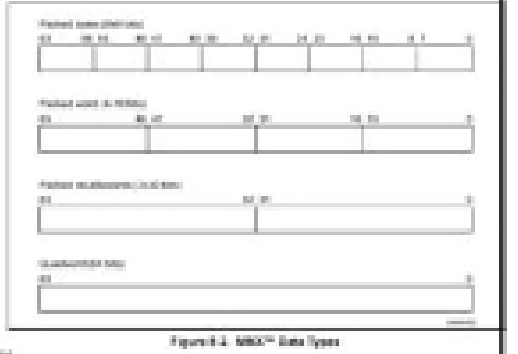


Figure 8-1. MMX™ Register Set

Saturating Arithmetic

The MMX instructions perform SIMD operations between MMX registers on packed bytes, words, or dwords.

The arithmetic operations can made to operate in Saturation mode.

What saturation mode does is clip numbers to Maximum positive or maximum negative values during arithmetic.

In normal mode: FFh + 01h = 00h (unsigned overflow)
In saturated, unsigned mode: FFh + 01 = FFh (saturated to maximum value, closer to actual arithmetic value)

In normal mode: 7fh + 01h = 80h (signed overflow)

In saturated, signed mode: 7fh + 01 = 7fh (saturated to max value)

Why Saturating Arithmetic?

- In case of integer overflow (either signed or unsigned), many applications are satisfied with just getting an answer that is close to the right answer or saturated to maxium result
- Many DSP (Digital Signal Processing) algorithms depend on this feature
  - Many DSP algorithms for audio data (8 to 16 bit data) and Video data (8-bit R,G,B values) are integer based, and need saturating arithmetic.
- This is easy to implement in hardware, but slow to emulate in software. A nice feature to have.