

Instruction Set Progression

from MMX™ Technology
through Streaming SIMD Extensions 2



This article summarizes the progression of change to the instruction set in the Intel IA-32 architecture, from MMX™ technology to Streaming SIMD Extensions (SSE) to Streaming SIMD Extensions 2 (SSE2). The discussion begins with the MMX instruction set in the Intel® Pentium® II processor and progresses through to the SSE2 instruction set in the Intel® Pentium® 4 processor/ Foster processor.

The following table identifies the instruction sets support for each of the processors.

Processors	Supported Instruction Sets		
	MMX	SSE	SSE2
Intel® Pentium® Pro Processor			
Intel® Pentium® II Processor	x		
Intel® Pentium® II Xeon™ Processor	x		
Intel® Pentium® III Processor	x	x	
Intel® Pentium® III Xeon™ Processor	x	x	
Intel® Pentium® 4 Processor	x	x	x
Foster Processor	x	x	x

1.0 Intel® Pentium® Pro Processor to Intel® Pentium® II Processor

The major architectural modification between the Intel Pentium Pro processor and the Intel Pentium II processor was the addition of the MMX instruction set. The MMX instructions allow execution of the same instruction (single instruction stream) in parallel on different data sets (multiple data streams).

The MMX instruction set consists of 57 instructions that use eight 64-bit registers, MM0-MM7 aliased with the floating-point register stack. Note that the MMX registers MM0 through MM7 are mapped to the floating-point registers R0 through R7, respectively, not to the relative locations of the registers in the stack (ST0 through ST7). Each MMX register is mapped to the lowest 64 bits of the corresponding floating-point register.

The content of an MMX register can be eight bytes, four words, two double words, or a quad word. This introduced four new data types: packed bytes, packed words, packed double words and quad word.

Key attributes of the MMX instruction set are:

- All MMX instructions are integer instructions.
- Only data-transfer instructions can have a memory operand as the destination.
- All non-data-transfer instructions must have an MMX register as the destination. The source can be either an MMX register or a memory location.
- The mnemonic for every non-data-transfer instruction is prefixed with 'P' for packed.
- MMX instructions do not set flags.

The following subsections summarize the different types of MMX instructions.

1.1 Data Transfer

There are two data transfer instructions: MOVD and MOVQ. MOVQ transfers a quad word between two MMX registers or between an MMX register and memory. MOVD transfers a double word from an MMX register to memory and vice versa, or from an MMX register to an integer register and vice versa. When the destination operand is an MMX register, the operation writes the 32-bit source value to the lower 32 bits of the MMX register and zero extends the value to 64 bits.

1.2 Arithmetic

MMX technology supports a new arithmetic capability known as saturation arithmetic. When an overflow or underflow occurs under saturation arithmetic, instead of wrapping around the result, the operation sets the results to the maximum or minimum for the range.

The different types of arithmetic instructions are:

- Packed Add and Subtract: Operands for these instructions can be bytes, words or double words. The instruction set includes signed and unsigned saturation versions of the instructions.

- **Packed Multiply:** This instruction supports word operands only. The operation can retrieve the low words or the high words of the results on the destination register.
- **Packed Multiply Add:** This instruction supports word operands only. The operation adds the four product terms to get the 32-bit result.

1.3 Comparison

The MMX instruction set provides two forms of instructions for comparisons: PCMPEQB/W/D and PCMPGTB/W/D. PCMPEQB/W/D instructions compare for an equal to condition. PCMPGTB/W/D instructions compare for a greater than condition. Operands for these instructions can be bytes, words, or double words, as specified by the instruction suffix B, W, or D, respectively. Each operation writes masks of all 1's or all 0's to the destination MMX register at the granularity specified by the instruction suffix. As with all MMX instructions, these instructions do not set flags.

1.4. Conversion

Instructions are available to perform the following conversions:

- Packed bytes to packed words and vice versa (PACKSSWB, PACKUSWB, PUNPCKHBW and PUNPCKLBW).
- Packed double words to words and vice versa (PACKSSDW, PUNPCKHWD and PUNPCKLWD).
- Double word to quad word (PUNPCKHDQ and PUNCPKLDQ).

The following attributes apply to the conversion instructions:

- During packing (move from higher data width to a lower data width), saturation occurs.
- Packing of words to bytes has both signed and unsigned saturation.
- Packing of double words to words has signed saturation only.

1.5 Logical

The logical operations are PAND, POR, PXOR and PANDN, which perform on 64-bit quantities. These are bit-wise operations. The destination operand is an MMX register. The other operand can be an MMX register or memory.

1.6 Shift

Logical shifts are for left and right shifts on words, double words and quad words (PSL/RLW, PSL/Rd and PSL/RQ). Arithmetic shifts are for right shift only on words and double words (PSRAW and PSRAD). The shifts occur within each word, double word or quad word, according to the specifier of the suffix, W, D or Q, respectively.

1.7 Empty MMX state

The EMMS instruction empties the MMX state, setting the values of all the tags in the FPU tag word to empty (all 1's). This is required if the x87 FP instructions are to be used after using MMX instructions because the MMX registers share the x87 FP register stack.

2.0 Intel Pentium II Processor to Intel Pentium III Processor

The Pentium III processor has 70 more instructions than the Pentium II processor, called Streaming SIMD Extensions (SSE). The new SSE instructions use a new set of eight 128-bit registers, called XMM registers. Since these registers are physically different from the existing registers, a new machine state is added. The SSE instructions extend the SIMD architectural concept by widening the data stream and by adding single precision Floating-Point (FP) instructions as well.

The SSE instruction set also includes the scalar versions of SIMD instructions. While packed instructions operate on each pair of operands, scalar instructions operate only on the least significant pair (the others are left intact). If a memory operand appears on a non-data-transfer type of instruction, that operand must be 16-byte aligned; otherwise, a GP exception occurs.

Key attributes of the SSE instruction set are:

- Excluding additional SIMD integer, state management and cache control instructions, all other instructions are on single precision FP operands (32 bits) and always use 128-bit XMM registers.
- Additional SIMD integer instructions are extensions to MMX instructions; they use MMX registers and mnemonics use 'P' prefix for packed. None of the other instructions use the 'P' prefix.
- Data Transfer, Arithmetic, Comparison and Shuffle instructions use the suffixes SS (Scalar Single precision) and PS (Packed Single precision).

The following subsections summarize different types of instructions in the SSE set.

2.1 Data Transfer

The instructions in this category are MOVAPS, MOVUPS, MOVHPS, MOVLPS, MOVHLPS, MOVLHPS, MOVSS and MOVMSKPS. MOVAPS and MOVUPS transfer 16-bytes of data between two XMM registers or between an XMM register and memory. For MOVAPS, memory must be 16-

byte aligned. MOVHPS (MOVLPS) transfers eight bytes of data from memory to the upper (lower) two fields of an XMM register or vice versa. MOVHLPS (MOVLHPS) transfers eight bytes of data from the upper (lower) two fields of an XMM register to the lower (upper) two fields of an XMM register. MOVSS transfers four bytes of data between the lowest fields of two XMM registers or between memory and an XMM register. MOVMSKPS transfers the most significant bit of each of the four operands of an XMM register to an IA integer register.

2.2 Arithmetic

The arithmetic instructions perform add, subtract, multiply, divide, square root, maximum and minimum. For each operation, there are two instructions: one packed and one scalar. For example, the two multiply instructions are MULPS (multiply packed single precision) and MULSS (multiply scalar single precision).

2.3 Comparison

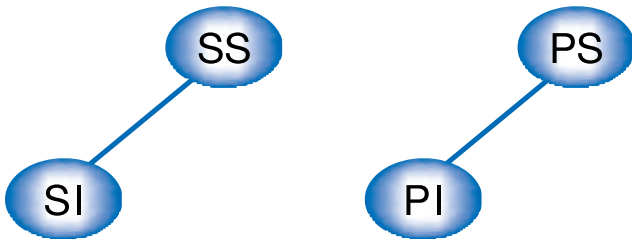
In MMX, comparison is only for equal to or greater than conditions. In contrast, SSE supports a full set of 12 comparison conditions. Unlike MMX (where the condition is part of the opcode), SSE uses a third operand to indicate the condition to be tested. The two comparison instructions are CMPPS (vector) and CMPSS (scalar). There are two other scalar comparison instructions, COMISS and UCOMISS, that set the flags in EFLAGS register according to the result.

2.4 Conversion

The conversions are between data types Packed Integer (PI) and Packed Single Precision FP (PS), or between Scalar Integer (SI) and Scalar Single Precision FP (SS). SI resides in a general-purpose register or a 32-bit memory operand. PI resides in an MMX register or a 64-bit memory operand. SS and PS reside in an XMM register or a 128-bit memory operand.

The following figure shows the conversions available. Each line shows conversion in either direction. When the conversion is from FP to integer (from SS to SI or from PS

to PI), the result can be rounded according to the bits set in MXCSR register (mnemonic prefix CVT) or truncated (mnemonic prefix CVTT). Therefore, the figure contains six conversion instructions.



2.5 Logical

The SSE logical operations are similar to the operations in MMX but operate on 128 bits.

2.6 Additional SIMD integer instructions

As stated before, these instructions are extensions to the MMX instruction set. They are all packed instructions. PMOVMSKB is similar to MOVMSKPS except that it considers eight bytes in an MMX register instead of four single precision FP operands in an XMM register.

The MMX instruction set has 16-bit signed multiply instructions to return lower halves of the products or the upper halves of the products. SSE introduces a 16-bit unsigned multiply instruction, PMULHUW, that returns the upper halves of the products. If the lower halves are needed, use PMULLW since the lower halves are the same whether the operands are signed or unsigned.

There is a new integer instruction, PSHUFW, which is different from other shuffle instructions (Section 2.7). Instruction “PSHUFW mm1, mm2/m64, imm8,” shuffles four words of mm2/m64 and places them in mm1 according to the imm8 value. Initial values of mm1 are unused.

2.7 Shuffle

SHUFPS uses two 128-bit operands and an immediate operand. The lower two FP operands of the destination are from the first operand of the instruction (XMM register) and the higher two FP operands of the destination are from the second operand of the instruction (XMM register or 128-bit memory).

2.8 State Management

The Pentium III processor has a new control/status register MXCSR (32-bit) used to mask and unmask numerical exceptions, set rounding modes, etc. The register can be loaded from 32-bit memory operand using the LDMXCSR instruction or stored to a 32-bit memory operand using the STMXCSR instruction. FXRSTOR automatically loads MXCSR. Similarly FXSAVE automatically saves MXCSR.

2.9 Cacheability Control

The cacheability control instructions introduced in SSE are MASKMOVQ, MOVNTQ, MOVNTPS, SFENCE, PREFETCHT0, PREFETCHT1, PREFETCHT2 and PREFETCHNTA.

SFENCE guarantees that all stores that precede SFENCE are globally visible before any store instructions that follow SFENCE are globally visible.

MOVNTPS writes four Single-Precision FP operands in an XMM register to memory, bypassing the caches, unless the line is already in the cache. Similarly, MOVNTQ writes the content of an MMX register to memory directly, unless the line is already in cache. MASKMOVQ also performs direct writing to the memory. These three non-temporal move instructions are weakly ordered, and therefore, SFENCE instruction may be required in MP systems.

PREFETCH0 prefetches data into all cache levels. PREFETCH1(2) prefetches data into level 1 (2) and higher level caches. PREFETCHNTA prefetches data to cache closest to the processor.

3.0 Intel Pentium III Processor to Intel Pentium 4 Processor

The Willamette processor introduces a set of 144 new instructions called Streaming SIMD Extensions 2 (SSE2). It uses the 128-bit XMM registers for SIMD operations. It does not introduce a new state to the machine. FXSAVE and FXRSTOR will take care of the x87-FP, MMX, and SSE states.

It also introduces two new data types: packed double precision FP and 128-bit integer. The 128-bit integer is called a Double Quad word (DQ), and is never used as an operand in an arithmetic instruction. The new instructions can be categorized as packed and scalar double precision FP operations, conversions, 128-bit MMX technology enhancements, and cacheability enhancement instructions. The most notable is the addition of SIMD double precision FP instructions.

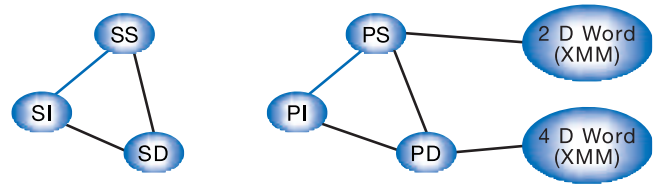
The following subsections summarize the different classes of instructions.

3.1 SIMD Double Precision FP

For every SIMD single precision FP instruction in SSE, there is a corresponding SIMD double precision FP instruction in SSE2, except for the reciprocal functions RCPPS, RCPSS, RSQRTPS and RSQRTSS.

3.2 Conversion

In addition to the four data types SSE used for conversion, namely SI, PI, SS and PS, SSE2 has two new types, Scalar Double Precision (SD) and Packed Double Precision (PD). These new data types reside in an XMM register or a 128-bit memory operand. The following figure shows all the conversion instructions. Those from SSE are shown in blue. Note that each line represents conversions in either direction. When the conversion is to an integer data type, the operation rounds the result according to the rounding bits in the MXCSR register (mnemonic prefix CVT) or truncates the result (mnemonic prefix CVTT).



3.3 128-bit Integer MMX Technology Enhancements

In SSE2, every MMX instruction, except the EMMS instruction, is extended to 128 bits by implementing the same functionality on a wider data format. Every additional SIMD integer instruction in SSE, except PSHUFW, has an extended version in SSE2. The PSHUFW instruction could not be extended to 128 bits because a 128-bit register contains eight words, and this operation needs eight bit-fields, where each bit-field can code a value between 0 and 7. This requires $8 \times 3 = 24$ bits which cannot be represented in imm8.

In addition to extending the existing MMX integer instructions to 128 bits, there are additional integer instructions:

- Move: MOVDQA (memory must be 16-byte aligned), MOVDQU, MOVDQ2Q and MOVQ2DQ.
- Arithmetic: PADDQ and PSUBQ
- Shuffle: PSHUFD, PSHUFW and PSHUFLW
- Shift: PSLLDQ and PSRLDQ
- Unpack: PUNPCKHQDQ and PUNPCKLQDQ

3.4 Cacheability Enhancements

SSE2 introduces several additional instructions to control cache. CLFLUSH flushes the line containing a given 8-bit memory operand from all caches. Invalidation is broadcast through the coherency domain (to all caches in a multi-processor system containing the invalidated line). This instruction, unlike INVD and WBINVD, can be used at all privileged levels.

SFENCE in SSE is supplemented by LFENCE and MFENCE in SSE2. LFENCE guarantees that every load preceding this instruction becomes globally visible before any load that follows LFENCE. MFENCE is similar except that loads and stores are considered together.

Additional non-temporal move instructions are:

- MOVNTPD, MOVNTDQ: If a memory operand is used it must be 16-byte aligned.
- MASKMOVDQU: Similar to MASKMOVQ in SSE but uses an XMM register and 128 bits in memory. No alignment is necessary.
- MOVNTI: Moves the contents of a general-purpose register to memory without polluting the cache.
- PAUSE: This alerts the processor that a spin-wait loop follows and the processor will decrease the number of speculative loads pumped to the pipeline. This reduces the penalty when the loop exits and also saves power through reduced use of resources.

The illustration at right graphically represents the different groups of instructions introduced in each post Intel Pentium Pro architecture.

For more information, visit <http://developer.intel.com/IDS>

MMX =

- 64-bit, packed, SIMD, integer instructions
- Use MM0-MM7 registers (aliased with R0-R7)
- Mnemonics have prefix 'P' except MOVQ, MOVD

MMX_{SSE} = MMX +

- Additional MMX instructions in Pentium III
- PAVGB/W, PEXTRW, PINSRW, ..., etc.

MMX_{SSE} =

- Expand MMX_{SSE} to include 129-bit data format

SSE =

- Eight XMM registers (new processor state), SP FP instructions, Scalar versions of SIMD, MMX_{SSE1}, State management, Shuffle, General comparison, additional conversion, Cacheability control, streaming store

SSE2 =

- SIMD DP FP, new conversion
- Additional streaming store
- MMX_{SSE2}
- Additional cacheability control
- Pause

