Table 6-20. M2 MMX Instruction Set Clock Count Summary

MMX INSTRUCTIONS	OPCODE	OPERATION	CLOCK COUNT	NOTES
EMMS *Empty MMX State*	0F77	Set the FP tag word ---> empty.	1	
MOVD *Move Doubleword*		Move dword from MMX register ---> integer register/memory		
Register to MMX Register	0F6E [11 mm reg]	or from integer register/memory ---> MMX register.	1	
MMX Register to Register	0F7E [11 mm reg]		1	
Memory to MMX Register	0F6E [mod mm r/m]		1	
MMX Register to Memory	0F7E [mod mm r/m]		1	
MOVQ *Move Quardword*		Move qword from MMX register ---> MMX register/memory		
MMX Register 2 to MMX Register 1	0F6F [11 mm1 mm2]	or from memory ---> MMX register.	1	
MMX Register 1 to MMX Register 2	0F7F [11 mm1 mm2]		1	
Memory to MMX Register	0F6F [mod mm r/m]		1	
MMX Register to Memory	0F7F [mod mm r/m]		1	
PACKSSDW *Pack Dword with Signed Saturation*		Pack, saturate signed dwords from MMX register and MMX register/		
MMX Register 2 to MMX Register1	0F6B [11 mm1 mm2]	memory ---> signed words in MMX register.	1	
Memory to MMX Register	0F6B [mod mm r/m]		1	
PACKSSWB *Pack Word with Signed Saturation*		Pack, saturate signed words from MMX register and MMX register/		
MMX Register 2 to MMX Register1	0F63 [11 mm1 mm2]	memory ---> signed bytes in MMX register.	1	
Memory to MMX Register	0F63 [mod mm r/m]		1	
PACKUSWB *Pack Word with Unsigned Saturation*		Pack, saturate signed words from MMX register and MMX register/		
MMX Register 2 to MMX Register1	0F67 [11 mm1 mm2]	memory ---> unsigned bytes in MMX register.	1	
Memory to MMX Register	0F67 [mod mm r/m]		1	
PADDB *Packed Add Byte with Wrap-Around*		Sum packed byte from MMX register/memory ---> packed byte in MMX		
MMX Register 2 to MMX Register1	0FFC [11 mm1 mm2]	register.	1	
Memory to MMX Register	0FFC [mod mm r/m]		1	
PADDD *Packed Add Dword with Wrap-Around*		Sum packed dword from MMX register/memory ---> packed dword in		
MMX Register 2 to MMX Register1	0FFE [11 mm1 mm2]	MMX register.	1	
Memory to MMX Register	0FFE [mod mm r/m]		1	
PADDSB *Packed Add Signed Byte with Saturation*		Sum signed packed byte from MMX register/memory ---> signed packed		
MMX Register 2 to MMX Register1	0FEC [11 mm1 mm2]	byte in MMX register and saturate.	1	
Memory to Register	0FEC [mod mm r/m]		1	
PADDSW *Packed Add Signed Word with Saturation*		Sum signed packed word from MMX register/memory ---> signed packed		
MMX Register 2 to MMX Register1	0FED [11 mm1 mm2]	word in MMX register and saturate.	1	
Memory to Register	0FED [mod mm r/m]		1	
PADDUSB *Add Unsigned Byte with Saturation*		Sum unsigned packed byte from MMX register/memory ---> unsigned		
MMX Register 2 to MMX Register1	0FDC [11 mm1 mm2]	packed byte in MMX register and saturate.	1	
Memory to Register	0FDC [mod mm r/m]		1	
PADDUSW *Add Unsigned Word with Saturation*		Sum unsigned packed word from MMX register/memory ---> unsigned		
MMX Register 2 to MMX Register1	0FDD [11 mm1 mm2]	packed word in MMX register and saturate.	1	
Memory to Register	0FDD [mod mm r/m]		1	
PADDW *Packed Add Word with Wrap-Around*		Sum packed word from the MMX register/memory ---> packed word in		
MMX Register 2 to MMX Register1	0FFD [11 mm1 mm2]	the MMX register.	1	
Memory to MMX Register	0FFD [mod mm r/m]		1	
PAND *Bitwise Logical AND*		AND 64 bits from the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0FDB [11 mm1 mm2]		1	
Memory to MMX Register	0FDB [mod mm r/m]		1	

\# = immediate 8-bit data

a = Cyrix Enhanced Instruction

Table 6-20. M2 MMX Instruction Set Clock Count Summary (Continued)

MMX INSTRUCTIONS	OPCODE	OPERATION	CLOCK COUNT	NOTES
PANDN *Bitwise Logical AND NOT* MMX Register 2 to MMX Register1 Memory to MMX Register	 0FDF [11 mm1 mm2] 0FDF [mod mm r/m]	Invert the 64 bits in the MMX register, AND inverted the MMX register with the MMX register/memory.	 1 1	
PCMPEQB *Packed Byte Compare for Equality* MMX Register 2 with MMX Register1 Memory with MMX Register	 0F74 [11 mm1 mm2] 0F74 [mod mm r/m]	Compare packed byte in the MMX register/memory with packed byte in the MMX register for equality.	 1 1	
PCMPEQD *Packed Dword Compare for Equality* MMX Register 2 with MMX Register1 Memory with MMX Register	 0F76 [11 mm1 mm2] 0F76 [mod mm r/m]	Compare packed dword in the MMX register/memory with packed dword in the MMX register for equality.	 1 1	
PCMPEQW *Packed Word Compare for Equality* MMX Register 2 with MMX Register1 Memory with MMX Register	 0F75 [11 mm1 mm2] 0F75 [mod mm r/m]	Compare packed word in the MMX register/memory with packed word in the MMX register for equality.	 1 1	
PCMPGTB *Pack Compare Greater Than Byte* MMX Register 2 to MMX Register1 Memory with MMX Register	 0F64 [11 mm1 mm2] 0F64 [mod mm r/m]	Compare packed byte in the MMX register with packed byte in the MMX register/memory for greater value.	 1 1	
PCMPGTD *Pack Compare Greater Than Dword* MMX Register 2 to MMX Register1 Memory with MMX Register	 0F66 [11 mm1 mm2] 0F66 [mod mm r/m]	Compare packed dword in the MMX register with packed dword in the MMX register/memory for greater value.	 1 1	
PCMPGTW *Pack Compare Greater Than Word* MMX Register 2 to MMX Register1 Memory with MMX Register	 0F65 [11 mm1 mm2] 0F65 [mod mm r/m]	Compare packed word in the MMX register with packed word in the MMX register/memory for greater value.	 1 1	
PMADDWD *Packed Multiply and Add* MMX Register 2 to MMX Register 1 Memory to MMX Register	 0FF5 [11 mm1 mm2] 0FF5 [mod mm r/m]	Multiply the packed word in the MMX register by the packed word in the MMX register/memory. Sum the 32-bit results pairwise and store in the MMX register as dword.	 2 2	
PMULHW *Packed Multiply High* MMX Register 2 to MMX Register1 Memory to MMX Register	 0FE5 [11 mm1 mm2] 0FE5 [mod mm r/m]	Multiply the signed packed word in the MMX register by the signed packed word in the MMX register/memory, then store the high-order 16-bits (bits 31-16) of the results in the MMX register.	 2 2	
PMULLW *Packed Multiply Low* MMX Register 2 to MMX Register1 Memory to MMX Register	 0FD5 [11 mm1 mm2] 0FD5 [mod mm r/m]	Multiply the packed word in the MMX register with the packed word in the MMX register/memory, then store the low-order 16 bits of the results in the MMX register.	 2 2	
POR *Bitwise OR* MMX Register 2 to MMX Register1 Memory to MMX Register	 0FEB [11 mm1 mm2] 0FEB [mod mm r/m]	OR 64 bits from the MMX register/memory with the MMX register.	 1 1	
PSLLD *Packed Shift Left Logical Dword* MMX Register 1 by MMX Register 2 MMX Register by Memory MMX Register by Immediate	 0FF2 [11 mm1 mm2] 0FF2 [mod mm r/m] 0F72 [11 110 mm] #	Shift dwords in the MMX register left by amount specified in the MMX register/memory/imm8, while shifting in zeros.	 1 1 1	
PSLLQ *Packed Shift Left Logical Qword* MMX Register 1 by MMX Register 2 MMX Register by Memory MMX Register by Immediate	 0FF3 [11 mm1 mm2] 0FF3 [mod mm r/m] 0F73 [11 110 mm] #	Shift the MMX register left by amount specified in the MMX register/ memory/imm8, while shifting in zeros.	 1 1 1	

\# = immediate 8-bit data

a = Cyrix Enhanced Instruction

Table 6-20. M2 MMX Instruction Set Clock Count Summary (Continued)

MMX INSTRUCTIONS	OPCODE	OPERATION	CLOCK COUNT	NOTES
PSLLW *Packed Shift Left Logical Word*		Shift words in the MMX register left by amount specified in the MMX register/memory/imm8, while shifting in zeros.		
MMX Register 1 by MMX Register 2	0FF1 [11 mm1 mm2]		1	
MMX Register by Memory	0FF1 [mod mm r/m]		1	
MMX Register by Immediate	0F71 [11 110mm] #		1	
PSRAD *Packed Shift Right Arithmetic* Dword		Shift dwords in the MMX register right by amount specified in the MMX register/memory/imm8, while shifting in sign bits.		
MMX Register 1 by MMX Register 2	0FE2 [11 mm1 mm2]		1	
MMX Register by Memory	0FE2 [mod mm r/m]		1	
MMX Register by Immediate	0F72 [11 100 mm]		1	
PSRAW *Packed Shift Right Arithmetic* Word		Shift words in the MMX register right by amount specified in the MMX register/memory/imm8, while shifting in sign bits.		
MMX Register 1 by MMX Register 2	0FE1 [11 mm1 mm2]		1	
MMX Register by Memory	0FE1 [mod mm r/m]		1	
MMX Register by Immediate	0F71 [11 100 mm]		1	
PSRLD *Packed Shift Right Logical Dword*		Shift dwords in the MMX register right by amount specified in the MMX register/memory/imm8, while shifting in zeros.		
MMX Register 1 by MMX Register 2	0FD2 [11 mm1 mm2]		1	
MMX Register by Memory	0FD2 [mod mm r/m]		1	
MMX Register by Immediate	0F72 [11 010 mm]		1	
PSRLQ *Packed Shift Right Logical Qword*		Shift words in the MMX register right by amount specified in the MMX register/memory/imm8, while shifting in zeros.		
MMX Register 1 by MMX Register 2	0FD3 [11 mm1 mm2]		1	
MMX Register by Memory	0FD3 [mod mm r/m]		1	
MMX Register by Immediate	0F73 [11 010 mm]		1	
PSRLW *Packed Shift Right Logical Word*		Shift words in the MMX register right by amount specified in the MMX register/memory/imm8, while shifting in zeros.		
MMX Register 1 by MMX Register 2	0FD1 [11 mm1 mm2]		1	
MMX Register by Memory	0FD1 [mod mm r/m]		1	
MMX Register by Immediate	0F71 [11 010 mm]		1	
PSUBB *Subtract Byte With Wrap-Around*		Subtract packed byte in the MMX register/memory from packed byte in the MMX register.		
MMX Register 2 to MMX Register1	0FF8 [11 mm1 mm2]		1	
Memory to MMX Register	0FF8 [mod mm r/m]		1	
PSUBD *Subtract Dword With Wrap-Around*		Subtract packed dword in the MMX register/memory from packed dword in the MMX register.		
MMX Register 2 to MMX Register1	0FFA [11 mm1 mm2]		1	
Memory to MMX Register	0FFA [mod mm r/m]		1	
PSUBSB *Subtract Byte Signed With Saturation*		Subtract signed packed byte in the MMX register/memory from packed byte in the MMX register and saturate.		
MMX Register 2 to MMX Register1	0FE8 [11 mm1 mm2]		1	
Memory to MMX Register	0FE8 [mod mm r/m]		1	
PSUBSW *Subtract Word Signed With Saturation*		Subtract signed packed word in the MMX register/memory from signed packed word in the MMX register and saturate.		
MMX Register 2 to MMX Register1	0FE9 [11 mm1 mm2]		1	
Memory to MMX Register	0FE9 [mod mm r/m]		1	
PSUBUSW *Subtract Word Unsigned With Saturation*		Subtract unsigned packed word in the MMX register/memory from unsigned packed word in the MMX register and saturate.		
MMX Register 2 to MMX Register1	0FD9 [11 mm1 mm2]		1	
Memory to MMX Register	0FD9 [11 mm reg]		1	

= immediate 8-bit data

a = Cyrix Enhanced Instruction

Table 6-20. M2 MMX Instruction Set Clock Count Summary (Continued)

MMX INSTRUCTIONS	OPCODE	OPERATION	CLOCK COUNT	NOTES
PSUBUSB *Subtract Byte Unsigned With Saturation*		Subtract unsigned packed byte in the MMX register/memory from unsigned packed byte in the MMX register and saturate.		
MMX Register 2 to MMX Register1	0FD8 [11 mm1 mm2]		1	
Memory to MMX Register	0FD8 [11 mm reg]		1	
PSUBW *Subtract Word With Wrap-Around*		Subtract packed word in the MMX register/memory from packed word in the MMX register.		
MMX Register 2 to MMX Register1	0FF9 [11 mm1 mm2]		1	
Memory to MMX Register	0FF9 [mod mm r/m]		1	
PUNPCKHBW *Unpack High Packed Byte Data to Packed Words*		Interleave bytes from the high halves of the MMX register and the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0F68 [11 mm1 mm2]		1	
Memory to MMX Register	0F68 [11 mm reg]		1	
PUNPCKHDQ *Unpack High Packed Dword Data to Qword*		Interleave dwords from the high halves of the MMX register and the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0F6A [11 mm1 mm2]		1	
Memory to MMX Register	0F6A [11 mm reg]		1	
PUNPCKHWD *Unpack High Packed Word Data to Packed Dwords*		Interleave words from the high halves of the MMX register and the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0F69 [11 mm1 mm2]		1	
Memory to MMX Register	0F69 [11 mm reg]		1	
PUNPCKLBW *Unpack Low Packed Byte Data to Packed Words*		Interleave bytes from the low halves of the MMX register and the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0F60 [11 mm1 mm2]		1	
Memory to MMX Register	0F60 [11 mm reg]		1	
PUNPCKLDQ *Unpack Low Packed Dword Data to Qword*		Interleave dwords from the high halves of the MMX register and the MMX register/memory ---> the MMX register.		
MMX Register 2 to MMX Register1	0F62 [11 mm1 mm2]		1	
Memory to MMX Register	0F62 [11 mm reg]		1	
PUNPCKLWD *Unpack Low Packed Word Data to Packed Dwords*		Interleave words from the high halves of the MMX register and the MMX register/memory ---> MMX register.		
MMX Register 2 to MMX Register1	0F61 [11 mm1 mm2]		1	
Memory to MMX Register	0F61 [11 mm reg]		1	
PXOR *Bitwise XOR*		XOR 64 bits from the MMX register/memory ---> the MMX register		
MMX Register 2 to MMX Register1	0FEF [11 mm1 mm2]		1	
Memory to MMX Register	0FEF [11 mm reg]		1	

\# = immediate 8-bit data

a = Cyrix Enhanced Instruction