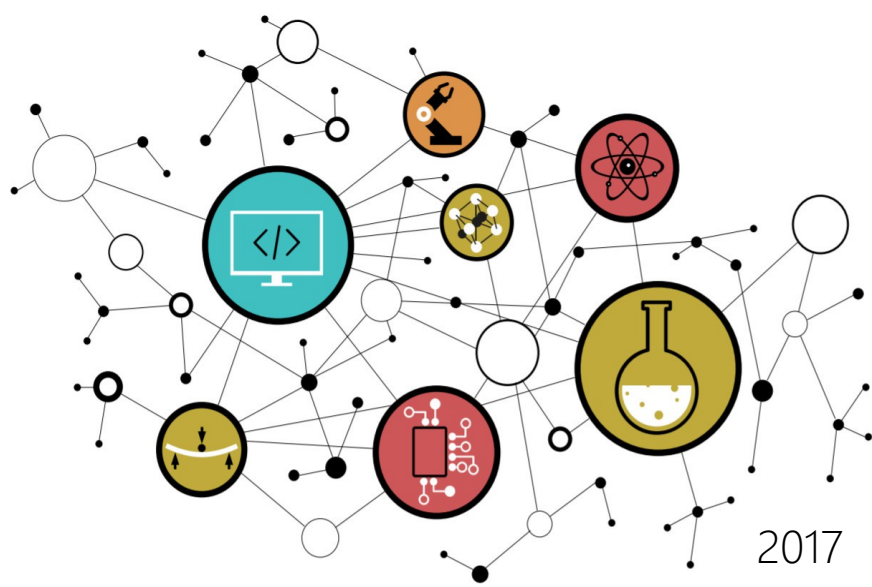


ANNEXES



SOMMAIRE

1. EXTRAITS – INTEL 64 AND IA-32 ARCHITECTURES – Software Developer's Manual – volume 1
2. JEU D'INSTRUCTIONS INTEL 8086
3. JEU D'INSTRUCTIONS INTEL SANDY BRIDGE – Agner Fog, Copenhagen University College of Engineering

GLOSSAIRE

1. EXTRAITS

Intel 64 and IA-32 Architectures – Software Developer's Manual – volume 1

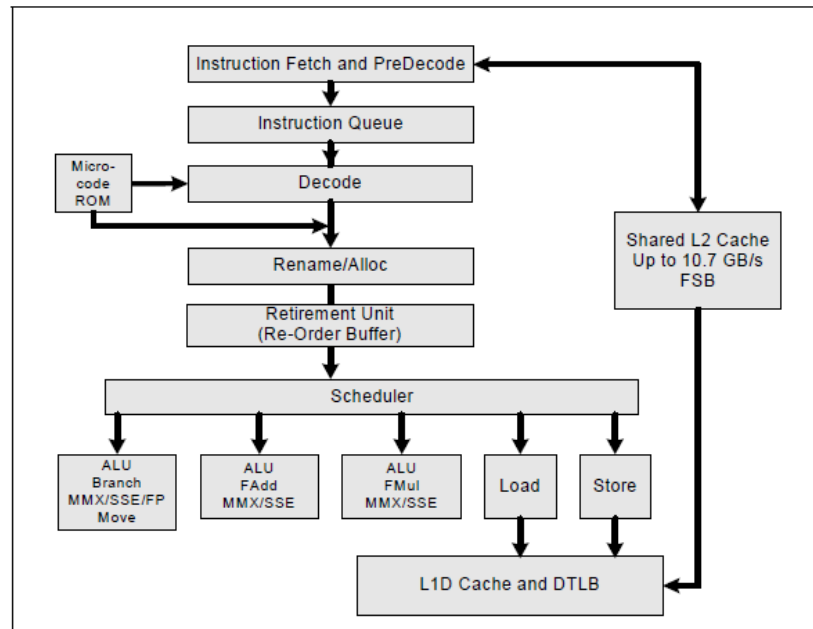


Figure 2-3. The Intel Core Microarchitecture Pipeline Functionality

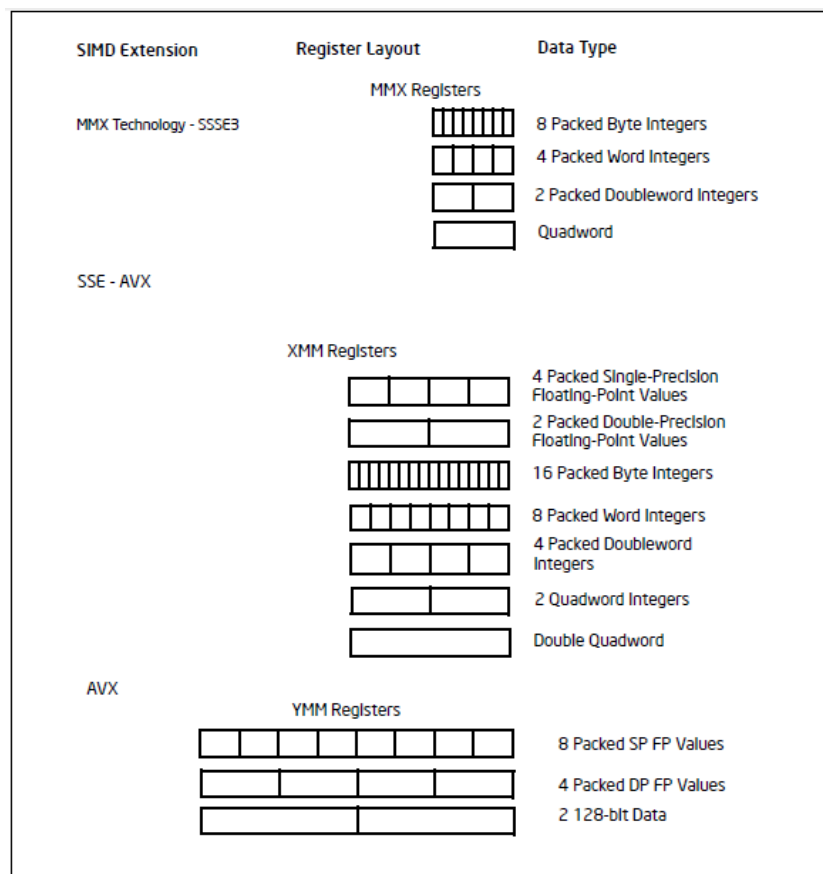


Figure 2-4. SIMD Extensions, Register Layouts, and Data Types

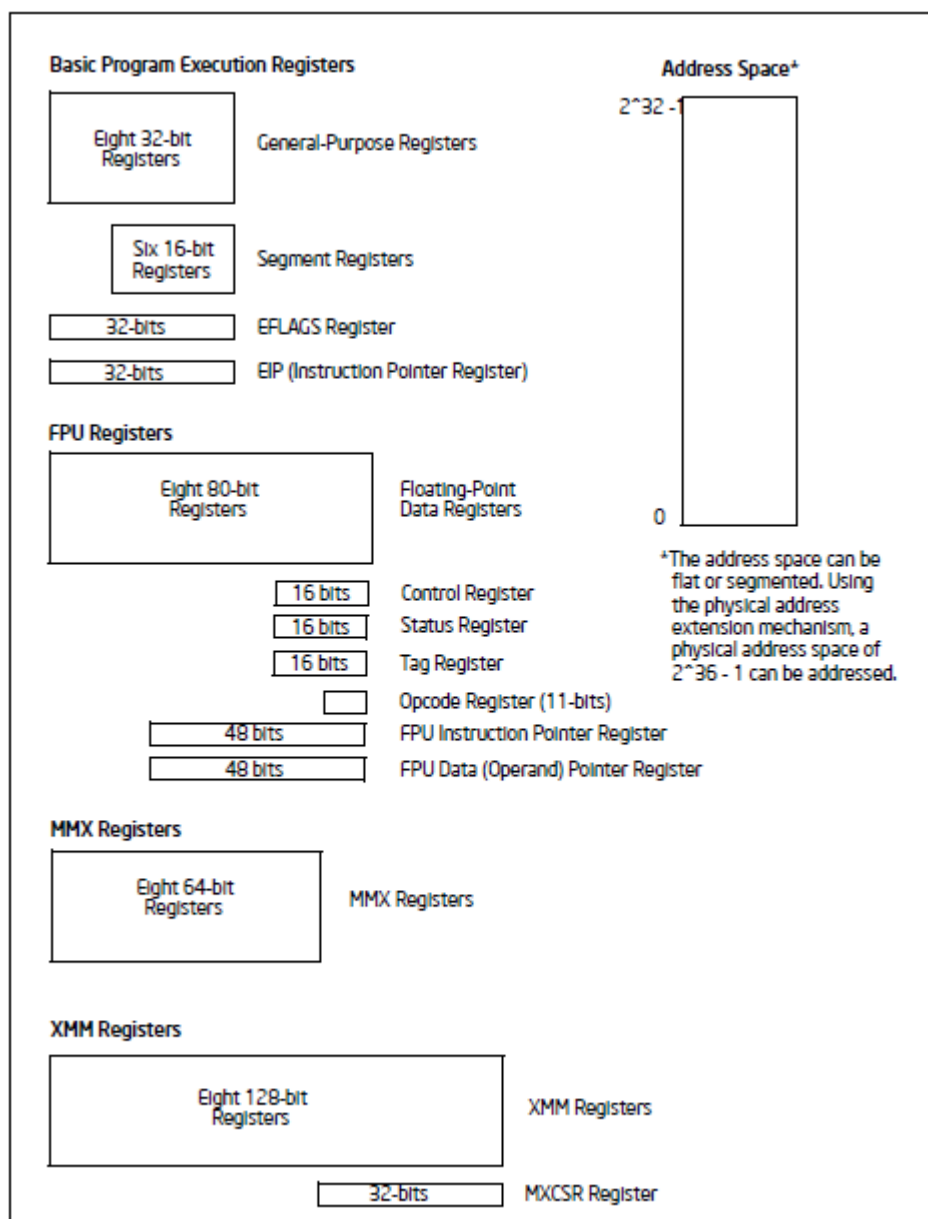


Figure 3-1. IA-32 Basic Execution Environment for Non-64-bit Modes

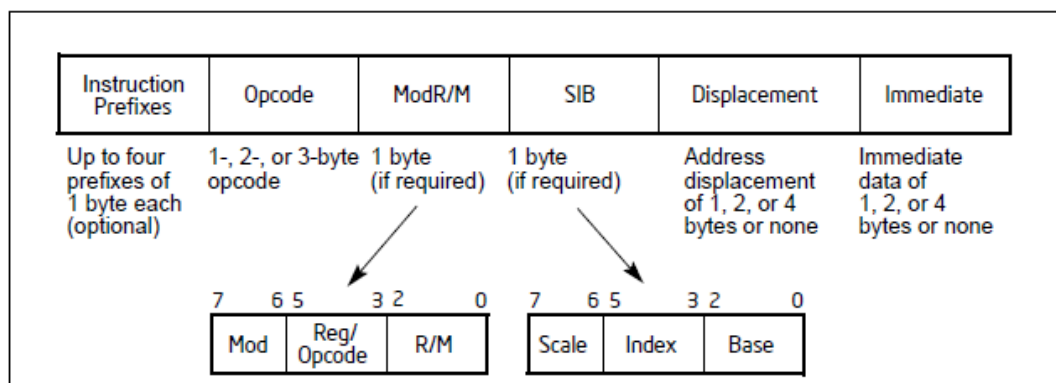


Figure 2-1. Intel 64 and IA-32 Architectures Instruction Format

2. JEU D'INSTRUCTIONS INTEL 8086

AAA	ASCII adjust AL after addition	HLT	Enter halt state
AAD	ASCII adjust AX before division	IDIV	Signed divide
AAM	ASCII adjust AX after multiplication	IMUL	Signed multiply
AAS	ASCII adjust AL after subtraction	IN	Input from port
ADC	Add with carry	INC	Increment by 1
ADD	Add	INT	Call to interrupt
AND	Logical AND	INTO	Call to interrupt if overflow
CALL	Call procedure	IRET	Return from interrupt
CBW	Convert byte to word	Jcc	Jump if condition
CLC	Clear carry flag	JMP	Jump
CLD	Clear direction flag	LAHF	Load flags into AH register
CLI	Clear interrupt flag	LDS	Load pointer using DS
CMC	Complement carry flag	LEA	Load Effective Address
CMP	Compare operands	LES	Load ES with pointer
CMPSB	Compare bytes in memory	LOCK	Assert BUS LOCK# signal
CMPSW	Compare words	LODSB	Load string byte
CWD	Convert word to doubleword	LODSW	Load string word
DAA	Decimal adjust AL after addition	LOOP/LOOPx	Loop control
DAS	Decimal adjust AL after subtraction	MOV	Move
DEC	Decrement by 1	MOVSB	Move byte from string to string
DIV	Unsigned divide	MOVSW	Move word from string to string
ESC	Used with floating-point unit	MUL	Unsigned multiply

NEG	<i>Two's complement negation</i>	SCASB	<i>Compare byte string</i>
NOP	<i>No operation</i>	SCASW	<i>Compare word string</i>
NOT	<i>Negate the operand, logical NOT</i>	SHL	<i>Shift left (unsigned shift left)</i>
OR	<i>Logical OR</i>	SHR	<i>Shift right (unsigned shift right)</i>
OUT	<i>Output to port</i>	STC	<i>Set carry flag</i>
POP	<i>Pop data from stack</i>	STD	<i>Set direction flag</i>
POPF	<i>Pop data from flags register</i>	STI	<i>Set interrupt flag</i>
PUSH	<i>Push data onto stack</i>	STOSB	<i>Store byte in string</i>
PUSHF	<i>Push flags onto stack</i>	STOSW	<i>Store word in string</i>
RCL	<i>Rotate left (with carry)</i>	SUB	<i>Subtraction</i>
RCR	<i>Rotate right (with carry)</i>	TEST	<i>Logical compare (AND)</i>
REPxx	<i>Repeat MOVs/STOS/CMPS/LODS/SCAS</i>	WAIT	<i>Wait until not busy</i>
RET	<i>Return from procedure</i>	XCHG	<i>Exchange data</i>
RETN	<i>Return from near procedure</i>	XLAT	<i>Table look-up translation</i>
RETF	<i>Return from far procedure</i>	XOR	<i>Exclusive OR</i>
ROL	<i>Rotate left</i>		
ROR	<i>Rotate right</i>		
SAHF	<i>Store AH into flags</i>		
SAL	<i>Shift Arithmetically left (signed shift left)</i>		
SAR	<i>Shift Arithmetically right (signed shift right)</i>		
SBB	<i>Subtraction with borrow</i>		

3. JEU D'INSTRUCTIONS INTEL SANDY BRIDGE

Intel Sandy Bridge

List of instruction timings and μ ops breakdown

Explanation of column headings:

Operands:	i = immediate data, r = register, mm = 64 bit mmx register, x = 128 bit xmm register, (x)mm = mmx or xmm register, y = 256 bit ymm register, same = same register for both operands. m = memory operand, m32 = 32-bit memory operand, etc.
μops fused domain:	The number of μ ops at the decode, rename, allocate and retirement stages in the pipeline. Fused μ ops count as one.
μops unfused domain:	The number of μ ops for each execution port. Fused μ ops count as two. Fused macro-ops count as one. The instruction has μ op fusion if the sum of the numbers listed under p015 + p23 + p4 exceeds the number listed under μ ops fused domain. A number indicated as '1+' under a read or write port means a 256-bit read or write operation using two clock cycles for handling 128 bits each cycle. The port cannot receive another read or write μ op in the second clock cycle, but a read port can receive an address-calculation μ op in the second clock cycle. An x under p0, p1 or p5 means that at least one of the μ ops listed under p015 can optionally go to this port. For example, a 1 under p015 and an x under p0 and p5 means one μ op which can go to either port 0 or port 5, whichever is vacant first. A value listed under p015 but nothing under p0, p1 and p5 means that it is not known which of the three ports these μ ops go to.
p015:	The total number of μ ops going to port 0, 1 and 5.
p0:	The number of μ ops going to port 0 (execution units).
p1:	The number of μ ops going to port 1 (execution units).
p5:	The number of μ ops going to port 5 (execution units).
p23:	The number of μ ops going to port 2 or 3 (memory read or address calculation).
p4:	The number of μ ops going to port 4 (memory write data).
Latency:	This is the delay that the instruction generates in a dependency chain. The numbers are minimum values. Cache misses, misalignment, and exceptions may increase the clock counts considerably. Where hyperthreading is enabled, the use of the same execution units in the other thread leads to inferior performance. Denormal numbers, NAN's and infinity do not increase the latency. The time unit used is core clock cycles, not the reference clock cycles given by the time stamp counter.
Reciprocal throughput:	The average number of core clock cycles per instruction for a series of independent instructions of the same kind in the same thread.
	The latencies and throughputs listed below for addition and multiplication using full size YMM registers are obtained only after a warm-up period of a thousand instructions or more. The latencies may be one or two clock cycles longer and the reciprocal throughputs double the values for shorter sequences of code. There is no warm-up effect when vectors are 128 bits wide or less.

Integer instructions

Instruction	Operands	μ ops fused domain	μ ops unfused domain					Latency	Reciprocal throughput	Comments
			p015	p0	p1	p5	p23	p4		
Move instructions										

MOV	r,ri	1	1	x	x	x		1	0.33	all addressing modes
MOV	r,m	1					1	2	0.5	
MOV	m,r	1					1	1	3	1
MOV	m,i	1					1	1	1	1
MOVNTI	m,r	2					1	1	~350	1
MOVSBX MOVZX	r,r	1	1	x	x	x			1	0.33
MOVSD										
MOVSBX MOVZX	r,m	1					1			0.5
MOVSD										
CMOVCc	r,r	2	2	x	x	x		2	1	
CMOVCc	r,m	2	2	x	x	x	1		1	
XCHG	r,r	3	3	x	x	x		2	1	
XCHG	r,m	8					2	1	25	implicit lock
XLAT		3	2				1		7	1
PUSH	r	1					1	1	3	1
PUSH	i	1					1	1		1
PUSH	m	2					2	1		1
PUSHF(D/Q)		3	2	x	x	x	1	1		1
PUSHA(D)		16	0				8	8		8
POP	r	1					1		2	0.5
POP	(E/R)SP	1	0				1			0.5
POP	m	2					2	1		1
POPF(D/Q)		9	8	x	x	x	1			18
POPA(D)		18	10				8			9
LAHF SAHF		1	1						1	1
SALC		3	3						1	1
LEA	r,m	1	1	1		1		1	0.5	not 64 bit simple
LEA	r,m	1	1		1			3	1	complex or rip relative
BSWAP	r32	1	1		1			1	1	
BSWAP	r64	2	2		2			2	1	
PREFETCHNTA	m	1					1			0.5
PREFETCHT0/1/2	m	1					1			0.5
LFENCE		2					1	1		4
MFENCE		3	1				1	1		33
SFENCE		2					1	1		6
Arithmetic instructions										
ADD SUB	r,ri	1	1	x	x	x		1	0.33	
ADD SUB	r,m	1	1	x	x	x	1		0.5	
ADD SUB	m,ri	2	1	x	x	x	2	1	6	1
SUB	r,same	1	0						0	0.25
ADC SBB	r,ri	2	2	x	x	x		2	1	
ADC SBB	r,m	2	2	x	x	x	1	2	1	
ADC SBB	m,ri	4	3	x	x	x	2	1	7	1.5
CMP	r,ri	1	1	x	x	x		1	0.33	
CMP	m,ri	1	1	x	x	x	1	1	0.5	

Sandy Bridge										
INC DEC NEG NOT	r	1	1	x	x	x		1	0.33	
INC DEC NEG NOT	m	3	1	x	x	x	2	1	6	2
AAA AAS		2	2						4	not 64 bit
DAA DAS		3	3						4	not 64 bit
AAD		3	3						2	not 64 bit
AAM		8	8						20	not 64 bit
MUL IMUL	r8	1	1		1				3	1
MUL IMUL	r16	4	4						4	2
MUL IMUL	r32	3	3						4	2
MUL IMUL	r64	2	2						3	1
IMUL	r,r	1	1		1				3	1
IMUL	r16,r16,i	2	2						4	1
IMUL	r32,r32,i	1	1		1				3	1
IMUL	r64,r64,i	1	1		1				3	1
MUL IMUL	m8	1	1		1		1		3	1
MUL IMUL	m16	4	3				1			2
MUL IMUL	m32	3	2				1			2
MUL IMUL	m64	2	1				1			2
IMUL	r,m	1	1		1		1			1
IMUL	r16,m16,i	2	2				1			1
IMUL	r32,m32,i	1	1		1		1			1
IMUL	r64,m64,i	1	1		1		1			1
DIV	r8	10	10						20-24	11-14
DIV	r16	11	11						21-25	11-14
DIV	r32	10	10						20-28	11-18
DIV	r64	34-56							30-94	22-76
IDIV	r8	10	10						21-24	11-14
IDIV	r16	10	10						21-25	11-14
IDIV	r32	9	9						20-27	11-18
IDIV	r64	59-138							40-103	25-84
CBW		1	1						1	0.5
CWDE		1	1			1			1	1
CDQE		1	1						1	0.5
CWD		2	2						1	1
CDQ		1	1						1	1
CQO		1	1						1	0.5
POPCNT	r,r	1	1		1				3	1
POPCNT	r,m	1	1		1		1			1
CRC32	r,r	1	1		1				3	1
CRC32	r,m	1	1		1		1			1
Logic instructions										
AND OR XOR	r,r/i	1	1	x	x	x			1	0.33
AND OR XOR	r,m	1	1	x	x	x	1			0.5
AND OR XOR	m,r/i	2	1	x	x	x	2	1	6	1
XOR	r,same	1	0						0	0.25
TEST	r,r/i	1	1	x	x	x			1	0.33
TEST	m,r/i	1	1	x	x	x	1			0.5
SHR SHL SAR	r,i	1	1	x		x			1	0.5

Sandy Bridge									
SHR SHL SAR	m,i	3	1			2	1	1	2
SHR SHL SAR	r,d	3	3					2	2
SHR SHL SAR	m,d	5	3			2	1		4
ROR ROL	r,i	1	1					1	1
ROR ROL	m,i	4	3			2	1		2
ROR ROL	r,d	3	3					2	2
ROR ROL	m,d	5	3			2	1		4
RCR	r8,1	high						high	high
RCR	r16/32/64,1	3	3					2	2
RCR	r,i	8	8					5	5
RCR	m,i	11	7						6
RCR	r,d	8	8					5	5
RCR	m,d	11	7						6
RCL	r,i	3	3					2	2
RCL	r,i	8	8					6	6
RCL	m,i	11	7						6
RCL	r,d	8	8					6	6
RCL	m,d	11	7						6
SHRD SHLD	r,r,i	1	1						0.5
SHRD SHLD	m,r,i	3				2	1		2
SHRD SHLD	r,r,d	4	4					2	2
SHRD SHLD	m,r,d	5	3			2	1		4
BT	r,ri	1	1					1	0.5
BT	m,r	10	8			1			5
BT	m,i	2	1			1			0.5
BTR BTS BTC	r,ri	1	1					1	0.5
BTR BTS BTC	m,r	11	7			2	1		5
BTR BTS BTC	m,i	3	1			2	1		2
BSF BSR	r,r	1	1					3	1
BSF BSR	r,m	1	1		1	1			1
SETcc	r	1	1	x		x		1	0.5
SETcc	m	2	1	x		x	1	1	1
CLC		1	0						0.25
STC CMC		1	1	x	x	x		1	0.33
CLD STD		3	3						4
Control transfer instructions									
JMP	short/near	1	1			1		0	2
JMP	r	1	1			1		0	2
JMP	m	1	1			1	1	0	2
Conditional jump	short/near	1	1			1		0	1-2
Fused arithmetic and branch		1	1			1		0	1-2
J(E/R)CXZ	short	2	2	x	x	1			2-4
LOOP	short	7	7						5
LOOP(N)E	short	11	11						5
CALL	near	3	2			1	1	1	2
CALL	r	2	1			1	1	1	2

fastest if
not jump-
ing

Sandy Bridge									
FBSTP	m80	246						252	
FXCH	r	1	0				0	0.5	
FILD	m	1	1	1	1		6	1	
FIST(P)	m	3	1	1	1	1	7	2	
FISTTP	m	3	1	1	1	1	7	2	SSE3
FLDZ		1	1	1				2	
FLD1		2	2	1	1			2	
FLDPI FLDL2E etc.		2	2	2				2	
FCMOVcc	r	3	3				3	2	
FNSTSW	AX	2	2				2	1	
FNSTSW	m16	2	1		1	1		1	
FLDCW	m16	3	2		1		8		
FNSTCW	m16	2	1	1	1	1	5	1	
FINCSTP FDECSTP		1	1	1			1	1	
FFREE(P)	r	1	1					1	
FNSAVE	m	143						166	
FRSTOR	m	90						165	
Arithmetic instructions									
FADD(P) FSUB(R)(P)	r	1	1	1			3	1	
FADD(P) FSUB(R)(P)	m	2	2	1	1			1	
FMUL(P)	r	1	1	1			5	1	
FMUL(P)	m	1	1	1	1			1	
FDIV(R)(P)	r	1	1	1			10-24	10-24	
FDIV(R)(P)	m	1	1	1	1			10-24	
FABS		1	1	1			1	1	
FCHS		1	1	1			1	1	
FCOM(P) FUCOM	r	1	1	1			3	1	
FCOM(P) FUCOM	m	1	1	1	1			1	
FCOMPP FUCOMPP		2	2	1	1			1	
FCOMI(P) FUCOMI(P)	r	3	3	1			4	1	
FIADD FISUB(R)	m	2	2	2	1			1	
FIMUL	m	2	2	1	1	1		1	
FIDIV(R)	m	2	2	1	1	1			
FICOM(P)	m	2	2	2	1			2	
FTST		1	1	1				1	
FXAM		2	2	1				2	
FPREM		28	28				21	21	
FPREM1		41-87					26-50	26-50	
FRNDINT		17	17				22		
Math									
FSCALE		27	27				12		
FXTRACT		17	17				10		
FSQRT		1	1	1			10-24		
FSIN		64-100					47-100		
FCOS		20-110					47-115		
FSINCOS		20-110					43-123		
F2XM1		53-118					61-69		
FYL2X		454	454				724		

Sandy Bridge

FYL2XP1	464	464					726	
FPTAN	102	102					130	
FPATAN	28-91						93-146	
Other								
FNOP	1	1	1					1
WAIT	2	2						1
FNCLEX	5	5						22
FNINIT	26	26						81

Integer MMX and XMM instructions

Instruction	Operands	μops fused domain	μops unfused domain						Latency	Reciprocal throughput	Comments
			p015	p0	p1	p5	p23	p4			
Move instructions											
MOVD	r32/64,(x)mm	1	1	x	x	x			1	0.33	
MOVD	m32/64,(x)mm	1					1	1	3	1	
MOVD	(x)mm,r32/64	1	1	x	x	x			1	0.33	
MOVD	(x)mm,m32/64	1					1		3	0.5	
MOVQ	(x)mm,(x)mm	1	1	x	x	x			1	0.33	
MOVQ	(x)mm,m64	1					1		1	0.5	
MOVQ	m64,(x)mm	1					1	1	3	1	
MOVDQA	x,x	1	1	x	x	x			1	0.33	
MOVDQA	x, m128	1					1		3	0.5	
MOVDQA	m128, x	1					1	1	3	1	
MOVDQU	x, m128	1	1				1		3	0.5	
MOVDQU	m128, x	1	1				1	1	3	1	
LDDQU	x, m128	1	1				1		3	0.5	SSE3
MOVDQ2Q	mm, x	2	2						1	1	
MOVQ2DQ	x,mm	1	1						1	0.33	
MOVNTQ	m64,mm	1					1	1	~300	1	
MOVNTDQ	m128,x	1					1	1	~300		
MOVNTDQA	x, m128	1					1			0.5	SSE4.1
PACKSSWB/DW PACKUSWB	mm,mm	1	1						1	1	
PACKSSWB/DW PACKUSWB	mm,m64	1	1		1		1				
PACKSSWB/DW PACKUSWB	x,x	1	1	x		x			1	0.5	
PACKSSWB/DW PACKUSWB	x,m128	1	1	x		x	1			0.5	
PACKUSDW	x,x	1	1	x		x			1	0.5	SSE4.1
PACKUSDW	x,m	1	1	x		x	1			0.5	SSE4.1
PUNPCKH/LBW/WD/DQ	(x)mm,(x)mm	1	1	x		x			1	0.5	
PUNPCKH/LBW/WD/DQ	(x)mm,m	1	1	x		x	1			0.5	
PUNPCKH/LQDQ	x,x	1	1	x		x			1	0.5	
PUNPCKH/LQDQ	x, m128	2	1	x		x	1			0.5	
PMOVSX/ZXBW	x,x	1	1	x		x			1	0.5	SSE4.1
PMOVSX/ZXBW	x,m64	1	1	x		x	1			0.5	SSE4.1

Sandy Bridge

PMOVSX/ZXBD	x,x	1	1	x	x		1	0.5	SSE4.1
PMOVSX/ZXBD	x,m32	1	1	x	x	1		0.5	SSE4.1
PMOVSX/ZXBQ	x,x	1	1	x	x		1	0.5	SSE4.1
PMOVSX/ZXBQ	x,m16	1	1	x	x	1		0.5	SSE4.1
PMOVSX/ZXWD	x,x	1	1	x	x		1	0.5	SSE4.1
PMOVSX/ZXWD	x,m64	1	1	x	x	1		0.5	SSE4.1
PMOVSX/ZXWQ	x,x	1	1	x	x		1	0.5	SSE4.1
PMOVSX/ZXWQ	x,m32	1	1	x	x	1		0.5	SSE4.1
PMOVSX/ZXDQ	x,x	1	1	x	x		1	0.5	SSE4.1
PMOVSX/ZXDQ	x,m64	1	1	x	x	1		0.5	SSE4.1
PSHUF8	(x)mm,(x)mm	1	1	x	x		1	0.5	SSSE3
PSHUF8	(x)mm,m	2	1	x	x	1		0.5	SSSE3
PSHUF8	mm,mm,i	1	1	x	x		1	0.5	
PSHUF8	mm,m64,i	2	1	x	x	1		0.5	
PSHUF8	xmm,x,i	1	1	x	x		1	0.5	
PSHUF8	x,m128,i	2	1	x	x	1		0.5	
PSHUF8/HW	x,x,i	1	1	x	x		1	0.5	
PSHUF8/HW	x, m128,i	2	1	x	x	1		0.5	
PALIGNR	(x)mm,(x)mm,i	1	1	x	x		1	0.5	SSSE3
PALIGNR	(x)mm,m,i	2	1	x	x	1		0.5	SSSE3
PBLENDB	x,x,xmm0	2	2	1	1		2	1	SSE4.1
PBLENDB	x,m,xmm0	3	2	1	1	1		1	SSE4.1
PBLENDB	x,x,i	1	1	x	x		1	0.5	SSE4.1
PBLENDB	x,m,i	2	1	x	x	1		0.5	SSE4.1
MASKMOVB	mm,mm	4	1	1		2	1	1	
MASKMOVBQ	x,x	10	4			4	x	6	
PMOVBQ	r32,(x)mm	1	1	1			2	1	
PEXTRB	r32,x,i	2	2	x	x		2	1	SSE4.1
PEXTRB	m8,x,i	2	1	x	x	1	1	1	SSE4.1
PEXTRW	r32,(x)mm,i	2	2	x	x		2	1	
PEXTRW	m16,(x)mm,i	2	1	x	x	1	1	2	SSE4.1
PEXTRD	r32,x,i	2	2	x	x		2	1	SSE4.1
PEXTRD	m32,x,i	3	2	x	x	1	1	1	SSE4.1
PEXTRQ	m64,x,i	2	2	x	x		2	1	SSE4.1, 64b
PEXTRQ	m64,x,i	3	2	x	x	1	1	1	
PINSRB	x,r32,i	2	2	x	x		2	1	SSE4.1
PINSRB	x,m8,i	2	1	x	x	1		0.5	SSE4.1
PINSRW	(x)mm,r32,i	2	2	x	x		2	1	
PINSRW	(x)mm,m16,i	2	1	x	x	1		0.5	
PINSRD	x,r32,i	2	2	x	x		2	1	SSE4.1
PINSRD	x,m32,i	2	1	x	x	1		0.5	SSE4.1
PINSRQ	x,r64,i	2	2	x	x		2	1	SSE4.1, 64 b
PINSRQ	x,m64,i	2	1	x	x	1		0.5	
Arithmetic instructions									
PADD/SUB(U,S)B/W/D/Q	(x)mm, (x)mm	1	1	x	x		1	0.5	
PADD/SUB(U,S)B/W/D/Q	(x)mm,m	1	1	x	x	1		0.5	
PHADD/SUB(S)W/D	(x)mm, (x)mm	3	3	x	x		2	1.5	SSSE3
PHADD/SUB(S)W/D	(x)mm,m64	4	3	x	x	1		1.5	SSSE3

Sandy Bridge

PCMPEQ/GB/W/D	(x)mm,m	1	1	x	x	1	0.5	
PCMPEQQ	x,x	1	1	x	x	1	0.5	SSE4.1
PCMPEQQ	x,m128	1	1	x	x	1	0.5	SSE4.1
PCMPGTQ	x,x	1	1	1		5	1	SSE4.2
PCMPGTQ	x,m128	1	1	1		1	1	SSE4.2
PSUBxx, PCMPGTx	x,same	1	0			0	0.25	
PCMPEQx	x,same	1	1			0	0.5	
PMULL/HW PMULHUW	(x)mm,(x)mm	1	1		1	5	1	
PMULL/HW PMULHUW	(x)mm,m	1	1		1	1	1	
PMULHRSW	(x)mm,(x)mm	1	1	1		5	1	SSSE3
PMULHRSW	(x)mm,m	1	1	1	1	1	1	SSSE3
PMULLD	x,x	1	1	1		5	1	SSE4.1
PMULLD	x,m128	2	1	1	1	1	1	SSE4.1
PMULDQ	x,x	1	1	1		5	1	SSE4.1
PMULDQ	x,m128	1	1	1	1	1	1	SSE4.1
PMULUDQ	(x)mm,(x)mm	1	1	1		5	1	
PMULUDQ	(x)mm,m	1	1	1	1	1	1	
PMADDWD	(x)mm,(x)mm	1	1	1		5	1	
PMADDWD	(x)mm,m	1	1	1	1	1	1	
PMADDUBSW	(x)mm,(x)mm	1	1	1		5	1	SSSE3
PMADDUBSW	(x)mm,m	1	1	1	1	1	1	SSSE3
PAVGB/W	(x)mm,(x)mm	1	1	x	x	1	0.5	
PAVGB/W	(x)mm,m	1	1	x	x	1	0.5	
PMIN/MAXSB	x,x	1	1	x	x	1	0.5	SSE4.1
PMIN/MAXSB	x,m128	1	1	x	x	1	0.5	SSE4.1
PMIN/MAXUB	(x)mm,(x)mm	1	1	x	x	1	0.5	
PMIN/MAXUB	(x)mm,m	1	1	x	x	1	0.5	
PMIN/MAXSW	(x)mm,(x)mm	1	1	x	x	1	0.5	
PMIN/MAXSW	(x)mm,m	1	1	x	x	1	0.5	
PMIN/MAXUW	x,x	1	1	x	x	1	0.5	SSE4.1
PMIN/MAXUW	x,m	1	1	x	x	1	0.5	SSE4.1
PMIN/MAXU/SD	x,x	1	1	x	x	1	0.5	SSE4.1
PMIN/MAXU/SD	x,m128	1	1	x	x	1	0.5	SSE4.1
PHMINPOSUW	x,x	1	1	1		5	1	SSE4.1
PHMINPOSUW	x,m128	1	1	1	1	1	1	SSE4.1
PABSB/W/D	(x)mm,(x)mm	1	1	x	x	1	0.5	SSSE3
PABSB/W/D	(x)mm,m	1	1	x	x	1	0.5	SSSE3
PSIGNB/W/D	(x)mm,(x)mm	1	1	x	x	1	0.5	SSSE3
PSIGNB/W/D	(x)mm,m	1	1	x	x	1	0.5	SSSE3
PSADBW	(x)mm,(x)mm	1	1	1		5	1	
PSADBW	(x)mm,m	1	1	1	1	1	1	
MPSADBW	x,x,i	3	3			6	1	SSE4.1
MPSADBW	x,m,i	4	3		1	1	1	SSE4.1
PCLMULQDQ	x,x,i	18	18			14	8	only in some processors
AESDEC, AESDECLAST, AESENC, AESENCCLAST								

Sandy Bridge

AESIMC	x,x	2	2					2	2	do.
AESKEYGENASSIST	x,x,i	11	11					8	8	do.
Logic instructions										
PAND(N) POR PXOR	(x)mm,(x)mm	1	1	x	x	x		1	0.33	
PAND(N) POR PXOR	(x)mm,m	1	1	x	x	x	1		0.5	
PXOR	x,same	1	0					0	0.25	
PTEST	x,x	1	1					1	1	SSE4.1
PTEST	x,m128	1	1				1		1	SSE4.1
PSLL/RL/RAW/D/Q	mm,mm/i	1	1		1			1	1	
PSLL/RL/RAW/D/Q	mm,m64	1	1		1		1		2	
PSLL/RL/RAW/D/Q	xmm,i	1	1		1			1	1	
PSLL/RL/RAW/D/Q	x,x	2	2					2	1	
PSLL/RL/RAW/D/Q	x,m128	3	2				1		1	
PSLL/RDQ	x,i	1	1					1	1	
String instructions										
PCMPESTR	x,x,i	8	8					4	4	SSE4.2
PCMPESTR	x,m128,i	8	7				1		4	SSE4.2
PCMPESTRM	x,x,i	8	8					11-12	4	SSE4.2
PCMPESTRM	x,m128,i	8	7				1		4	SSE4.2
PCMPISTR	x,x,i	3	3					3	3	SSE4.2
PCMPISTR	x,m128,i	4	3				1		3	SSE4.2
PCMPISTRM	x,x,i	3	3					11	3	SSE4.2
PCMPISTRM	x,m128,i	4	3				1		3	SSE4.2
Other										
EMMS		31	31						18	

Floating point XMM and YMM instructions

Instruction	Operands	μops fused domain	μops unfused domain						Latency	Reciprocal throughput	Comments
			p015	p0	p1	p5	p23	p4			
Move instructions											
MOVAPS/D	x,x	1	1			1			1	1	AVX
VMOVAPS/D	y,y	1	1			1			1	1	
MOVAPS/D MOVUPS/D	x,m128	1					1		3	0.5	
VMOVAPS/D											AVX
VMOVUPS/D	y,m256	1					1+		4	1	
MOVAPS/D MOVUPS/D	m128,x	1					1	1	3	1	
VMOVAPS/D											AVX
VMOVUPS/D	m256,y	1					1	1+	3	1	
MOVSS/D	x,x	1	1			1			1	1	
MOVSS/D	x,m32/64	1					1		3	0.5	
MOVSS/D	m32/64,x	1					1	1	3	1	
MOVHPS/D MOVLPD/D	x,m64	1	1			1	1		3	1	
MOVHLPD/D	m64,x	1	1			1	1	1	3	1	
MOVHPS/D MOVHPS	x,x	1	1			1			1	1	

Sandy Bridge										
MOVMSKPS/D	r32,x	1	1	1				2	1	
VMOVMSKPS/D	r32,y	1	1	1				2	1	
MOVNTPS/D	m128,x	1				1	1	~300	1	
VMOVNTPS/D	m256,y	1				1	4	~300	25	AVX
SHUFFPS/D	x,x,i	1	1		1			1	1	
SHUFFPS/D	x,m128,i	2	1		1	1			1	
VSHUFFPS/D	y,y,i	1	1		1			1	1	AVX
VSHUFFPS/D	y,y,m256,i	2	1		1	1+			1	AVX
VPERMILPS/PD	x,x,x/i	1	1		1			1	1	AVX
VPERMILPS/PD	y,y,y/i	1	1		1			1	1	AVX
VPERMILPS/PD	x,x,m	2	1		1	1			1	AVX
VPERMILPS/PD	y,y,m	2	1		1	1+			1	AVX
VPERMILPS/PD	x,m,i	2	1		1	1			1	AVX
VPERMILPS/PD	y,m,i	2	1		1	1+			1	AVX
VPERM2F128	y,y,y,i	1	1		1			2	1	AVX
VPERM2F128	y,y,m,i	2	1		1	1+			1	AVX
BLENDPS/PD	x,x,i	1	1		1			1	0.5	SSE4.1
BLENDPS/PD	x,m128,i	2	1		1	1			0.5	SSE4.1
VBLENDPS/PD	y,y,i	1	1		1			1	1	AVX
VBLENDPS/PD	y,m256,i	2	1		1	1+			1	AVX
BLENDVPS/PD	x,x,xmm0	2	2		2			2	1	SSE4.1
BLENDVPS/PD	x,m,xmm0	3	2		2	1			1	SSE4.1
VBLENDVPS/PD	y,y,y,y	2	2		2			2	1	AVX
VBLENDVPS/PD	y,y,m,y	3	2		2	1+			1	AVX
MOVDDUP	x,x	1	1		1			1	1	SSE3
MOVDDUP	x,m64	1				1		3	0.5	SSE3
VMOVDDUP	y,y	1	1		1			1	1	AVX
VMOVDDUP	y,m256	1				1+		3	1	AVX
VBROADCASTSS	x,m32	1				1			1	AVX
VBROADCASTSS	y,m32	2	1		1	1			1	AVX
VBROADCASTSD	y,m64	2	1		1	1			1	AVX
VBROADCASTF128	y,m128	2	1		1	1			1	AVX
MOVSH/LDUP	x,x	1	1		1			1	1	SSE3
MOVSH/LDUP	x,m128	1				1		3	0.5	SSE3
VMOVSH/LDUP	y,y	1	1		1			1	1	AVX
VMOVSH/LDUP	y,m256	1				1+		4	1	AVX
UNPCKH/LPS/D	x,x	1	1		1			1	1	SSE3
UNPCKH/LPS/D	x,m128	1	1		1	1			1	SSE3
VUNPCKH/LPS/D	y,y,y	1	1		1			1	1	AVX
VUNPCKH/LPS/D	y,y,m256	1	1		1	1+			1	AVX
EXTRACTPS	r32,x,i	2	2		1			2	1	SSE4.1
EXTRACTPS	m32,x,i	3	2		1	1	1		1	SSE4.1
VEXTRACTF128	x,y,i	1	1		1			2	1	AVX
VEXTRACTF128	m128,y,i	2	1			1	1		1	AVX
INSERTPS	x,x,i	1	1		1			1	1	SSE4.1
INSERTPS	x,m32,i	2	1		1	1			1	SSE4.1
VINSERTF128	y,y,x,i	1	1		1			2	1	AVX
VINSERTF128	y,y,m128,i	2	1		1	1			1	AVX
VMASKMOVPS/D	x,x,m128	3	2			1			1	AVX
VMASKMOVPS/D	y,y,m256	3	2			1+			1	AVX

Sandy Bridge										
VMASKMOVPS/D	m128,x,x	4	2			1	1		1	AVX
VMASKMOVPS/D	m256,y,y	4	2			1	1+		2	AVX
Conversion										
CVTPD2PS	x,x	2	2	1	1			3	1	
CVTPD2PS	x,m128	2	2	1	1				1	
VCVTPD2PS	x,y	2	2	1	1			4	1	AVX
VCVTPD2PS	x,m256	2	2	1	1+				1	AVX
CVTSD2SS	x,x	2	2	1	1			3	1	
CVTSD2SS	x,m64	2	2	1	1				1	
CVTPS2PD	x,x	2	2	1	1			3	1	
CVTPS2PD	x,m64	2	2	1	1	1			1	
VCVTPS2PD	y,x	2	2	1	1			4	1	AVX
VCVTPS2PD	y,m128	3	3		1	1			1	AVX
CVTSS2SD	x,x	2	2	1				3	1	
CVTSS2SD	x,m32	2	1	1		1			1	
CVTDQ2PS	x,x	1	1					3	1	
CVTDQ2PS	x,m128	1	1	1		1			1	
VCVTDQ2PS	y,y	1	1	1				3	1	AVX
VCVTDQ2PS	y,m256	1	1	1		1+			1	AVX
CVT(T) PS2DQ	x,x	1	1	1				3	1	
CVT(T) PS2DQ	x,m128	1	1	1		1			1	
VCVT(T) PS2DQ	y,y	1	1	1				3	1	AVX
VCVT(T) PS2DQ	y,m256	1	1	1		1+			1	AVX
CVTDQ2PD	x,x	2	2	1	1			4	1	
CVTDQ2PD	x,m64	2	2	1	1	1			1	
VCVTDQ2PD	y,x	2	2	1	1			5	1	AVX
VCVTDQ2PD	y,m128	3	2	1	1	1			1	AVX
CVT(T)PD2DQ	x,x	2	2	1	1			4	1	
CVT(T)PD2DQ	x,m128	2	2	1	1	1			1	
VCVT(T)PD2DQ	x,y	2	2	1	1			5	1	AVX
VCVT(T)PD2DQ	x,m256	2	2	1	1	1+			1	AVX
CVTPI2PS	x,mm	1	1	1				4	2	
CVTPI2PS	x,m64	1	1	1		1			2	
CVT(T)PS2PI	mm,x	2	2	1				4	1	
CVT(T)PS2PI	mm,m128	2	1	1		1			1	
CVTPI2PD	x,mm	2	2	1	1			4	1	
CVTPI2PD	x,m64	2	2	1	1	1			1	
CVT(T) PD2PI	mm,x	2	2					4	1	
CVT(T) PD2PI	mm,m128	2	2			1			1	
CVTSI2SS	x,r32	2	2	1				4	1.5	
CVTSI2SS	x,m32	1	1	1		1			1.5	
CVT(T)SS2SI	r32,x	2	2	1				4	1	
CVT(T)SS2SI	r32,m32	2	2	1		1			1	
CVTSI2SD	x,r32	2	2	1	1			4	1.5	
CVTSI2SD	x,m32	1	1	1		1			1.5	
CVT(T)SD2SI	r32,x	2	2	1				4	1	
CVT(T)SD2SI	r32,m64	2	2	1		1			1	

Sandy Bridge

ADDSS/D SUBSS/D	x,x	1	1	1		3	1	
ADDSS/D SUBSS/D	x,m32/64	1	1	1	1		1	
ADDPs/D SUBPs/D	x,x	1	1	1		3	1	
ADDPs/D SUBPs/D	x,m128	1	1	1	1		1	
VADDPs/D VSUBPs/D	y,y,y	1	1	1		3	1	AVX
VADDPs/D VSUBPs/D	y,y,m256	1	1	1	1+		1	AVX
ADDSUBPs/D	x,x	1	1	1		3	1	SSE3
ADDSUBPs/D	x,m128	1	1	1	1		1	SSE3
VADDSUBPs/D	y,y,y	1	1	1		3	1	AVX
VADDSUBPs/D	y,y,m256	1	1	1	1+		1	AVX
HADDPs/D HSUBPs/D	x,x	3	3	1	2	5	2	SSE3
HADDPs/D HSUBPs/D	x,m128	4	3	1	2	1	2	SSE3
VHADDPs/D								
VHSUBPs/D	y,y,y	3	3	1	2	5	2	AVX
VHADDPs/D								
VHSUBPs/D	y,y,m256	4	3	1	2	1+	2	AVX
MULSS MULPS	x,x	1	1	1		5	1	
MULSS MULPS	x,m	1	1	1		1	1	
VMULPS	y,y,y	1	1	1		5	1	AVX
VMULPS	y,y,m256	1	1	1	1+		1	AVX
MULSD MULPD	x,x	1	1	1		5	1	
MULSD MULPD	x,m	1	1	1	1		1	
VMULPD	y,y,y	1	1	1		5	1	AVX
VMULPD	y,y,m256	1	1	1	1+		1	AVX
DIVSS DIVPS	x,x	1	1	1		10-14	10-14	
DIVSS DIVPS	x,m	1	1	1	1		10-14	
VDIVPS	y,y,y	3	3	2	1	21-29	20-28	AVX
VDIVPS	y,y,m256	4	3	2	1	1+	20-28	AVX
DIVSD DIVPD	x,x	1	1	1		10-22	10-22	
DIVSD DIVPD	x,m	1	1	1	1		10-22	
VDIVPD	y,y,y	3	3	2	1	21-45	20-44	AVX
VDIVPD	y,y,m256	4	3	2	1+		20-44	AVX
RCPSS/PS	x,x	1	1	1		5	1	
RCPSS/PS	x,m128	1	1	1	1		1	
VRCPPS	y,y	2	3			7	2	AVX
VRCPPS	y,m256	4	3		1+		2	AVX
CMPccSS/D CMPccPS/D								
CMPccSS/D CMPccPS/D	x,x	1	1	1		3	1	
CMPccSS/D CMPccPS/D	x,m128	2	1	1	1		1	
VCMPccPS/D	y,y,y	1	1	1		3	1	AVX
VCMPccPS/D	y,y,m256	2	1	1	1+		1	AVX
COMISS/D UCOMISS/D	x,x	2	2			2	1	
COMISS/D UCOMISS/D	x,m32/64	2	2	1	1		1	
MAXSS/D MINSS/D	x,x	1	1	1		3	1	
MAXSS/D MINSS/D	x,m32/64	1	1	1	1		1	
MAXPS/D MINPS/D	x,x	1	1	1		3	1	
MAXPS/D MINPS/D	x,m128	1	1	1	1		1	
VMAXPS/D VMINPS/D	y,y,y	1	1	1		3	1	AVX
VMAXPS/D VMINPS/D	y,y,m256	1	1	1	1+		1	AVX

Sandy Bridge

ROUNDSS/SD/PS/PD	x,m128,i	2	1		1	1		1	SSE4.1
VROUNDSS/SD/PS/PD	y,y,i	1	1		1		3	1	AVX
VROUNDSS/SD/PS/PD	y,m256,i	2	1		1	1+		1	AVX
DPPS	x,x,i	4	4	1	2	1	12	2	SSE4.1
DPPS	x,m128,i	6	5			1		4	SSE4.1
VDPPS	y,y,y,i	4	4				12	2	AVX
VDPPS	y,m256,i	6	5			1+		4	AVX
DPPD	x,x,i	3	3				9	2	SSE4.1
DPPD	x,m128,i	4	3			1		2	SSE4.1
Math									
SQRTSS/PS	x,x	1	1	1			10-14	10-14	
SQRTSS/PS	x,m128	1	1	1		1		10-14	
VSQRTPS	y,y	3	3					21-28	AVX
VSQRTPS	y,m256	4	3			1+		21-28	AVX
SQRTSD/PD	x,x	1	1	1			10-21	10-21	
SQRTSD/PD	x,m128	2	1	1		1		10-21	
VSQRTPD	y,y	3	3				21-43	21-43	AVX
VSQRTPD	y,m256	4	3			1+		21-43	AVX
RSQRTPS/PS	x,x	1	1	1			5	1	
RSQRTPS/PS	x,m128	1	1	1		1		1	
VRSQRTPS	y,y	3	3				7	2	AVX
VRSQRTPS	y,m256	4	3			1+		2	AVX
Logic									
AND/ANDN/OR/XORPS/PD	x,x	1	1		1		1	1	
AND/ANDN/OR/XORPS/PD	x,m128	1	1		1	1		1	
VAND/ANDN/OR/XORPS/PD	y,y,y	1	1		1		1	1	AVX
VAND/ANDN/OR/XORPS/PD	y,y,m256	1	1		1	1+		1	AVX
(V)XORPS/PD	x/y,x/y,same	1	0				0	0.25	
Other									
VZEROUPPER		4					2	1	AVX
VZEROALL		12						11	AVX, 32 bit
VZEROALL		20						9	AVX, 64 bit
LDMXCSR	m32	3	3			1	1	3	
STMXCSR	m32	3	3		1	1	1	1	
VSTMXCSR	m32	3	3		1	1	1	1	AVX
FXSAVE	m4096	130						68	
FXRSTOR	m4096	116						72	
XSAVEOPT	m	100-161					60-500		

GLOSSAIRE

A

- **ABI** : Application Binary Interface
- **ADC** : Analog to Digital Converter
- **ALU** : Arithmetic and Logical Unit
- **AMD** : Advanced Micro Devices
- **ANSI** : American National Standards Institute
- **API** : Application Programming Interface
- **APU** : Accelerated Processor Unit
- **ARM** : société anglaise proposant des architectures CPU RISC 32bits
- **ASCII** : American Standard Code for Information Interchange

B

- **BP** : Base Pointer
- **BSL** : Board Support Library

C

- **CCS** : Code Composer Studio
- **CEM** : Compatibilité ElectroMagnétique
- **CISC** : Complex Instruction Set Computer
- **CPU** : Central Processing Unit
- **CSL** : Chip Support Library

D

- **DAC** : Digital to Analog Converter
- **DDR** : Double Data Rate
- **DDR SDRAM** : Double Data Rate Synchronous Dynamic Random Access Memory
- **DMA** : Direct Memory Access
- **DSP** : Digital Signal Processor
- **DSP** : Digital Signal Processing

E

- **EDMA** : Enhanced Direct Memory Access
- **EUSART** : Enhanced Universal Synchronous Asynchronous Receiver Transmitter
- **EMIF** : External Memory Interface
- **EPIC** : Explicitly Parallel Instruction Computing

F

- **FPU** : Floating Point Unit
- **FLOPS** : Floating-Point Operations Per Second
- **FMA**: Fused Multiply-Add

G

- **GCC** : Gnu Collection Compiler
- **GLCD** : Graphical Liquid Crystal Display
- **GNU** : GNU's Not UNIX
- **GPIO** : General Purpose Input Output
- **GPP** : General Purpose Processor
- **GPU** : Graphical Processing Unit

I

- **IA-64** : Intel Architecture 64bits
- **I2C** : Inter Integrated Circuit
- **ICC** : Intel C++ Compiler
- **IDE** : Integrated Development Environment
- **IDMA** : Internal Direct memory Access
- **IRQ** : Interrupt ReQuest
- **ISR** : Interrupt Software Routine
- **ISR** : Interrupt Service Routine

L

- **L1D** : Level 1 Data Memory
- **L1I** : Level 1 Instruction Memory (idem L1P)
- **L1P** : Level 1 Program Memory (idem L1I)
- **Lx** : Level x Memory
- **LCD** : Liquid Crystal Display
- **LRU** : Least Recently Used

M

- **MAC**: Multiply Accumulate
- **MCU** : Micro Controller Unit
- **MIMD** : Multiple Instructions on Multiple Data
- **MIPS** : Mega Instructions Per Second

- **MMU** : Memory Managment Unit
- **MPLABX** : MicrochiP LABoratory 10, IDE Microchip
- **MPU** : Micro Processor Unit ou GPP
- **MPU** : Memory Protect Unit

O

- **OS** : Operating System

P

- **PC** : Program Counter
- **PC** : Personal Computer
- **PIC18** : Famille MCU 8bits Microchip
- **PLD** : Programmable Logic Device
- **POSIX** : Portable Operating System Interface, héritage d'UNIX (norme IEEE 1003)
- **PPC** : Power PC

R

- **RAM** : Random Access Memory
- **RISC** : Reduced Instruction Set Computer
- **RS232** : Norme standardisant un protocole de communication série asynchrone
- **RTOS** : Real Time Operating System

S

- **SDK** : Software Development Kit
- **SIMD** : Single Instruction Multiple Date
- **SOB** : System On Board
- **SOC** : System On Chip
- **SOP** : Sums of products
- **SP** : Stack Pointer
- **SP** : Serial Port
- **SPI** : Serial Peripheral Interface
- **SRAM** : Static Random Access Memory
- **SSE** : Streaming SIMD Extensions
- **STM32** : STMicroelectronics 32bits MCU

T

- **TI** : Texas Instruments
- **TNS** : Traitement Numérique du Signal

- **TSC** : Time Stamp Counter
- **TTM** : Time To Market

U

- **UART** : Universal Asynchronous Receiver Transmitter
- **USB** : Universal Serial Bus

V

- **VHDL** : VHSIC Hardware Description langage
- **VHSIC** : Very High Speed Integrated Circuit
- **VLIW** : Very Long Intruction Word