http://sparksandflames.com/files/x86InstructionChart.html

Addressing Method Codes

A	Direct address. The instruction has no ModR/M byte; the address of the operand is encoded in the instruction; and no base register, index register, or scaling factor can be applied (for example, far JMP (EA)).
C	The reg field of the ModR/M byte selects a control register (for example, MOV (0F20, 0F22)).
D	The reg field of the ModR/M byte selects a debug register (for example, MOV (0F21,0F23)).
E	A ModR/M byte follows the opcode and specifies the operand. The operand is either a general-purpose register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register, an index register, a scaling factor, a displacement.
F	EFLAGS Register.
G	The reg field of the ModR/M byte selects a general register (for example, AX (000)).
I	Immediate data. The operand value is encoded in subsequent bytes of the instruction.
J	The instruction contains a relative offset to be added to the instruction pointer register (for example, JMP (0E9), LOOP).
M	The ModR/M byte may refer only to memory (for example, BOUND, LES, LDS, LSS, LFS, LGS, CMPXCHG8B).
O	The instruction has no ModR/M byte; the offset of the operand is coded as a word or double word (depending on address size attribute) in the instruction. No base register, index register, or scaling factor can be applied (for example, MOV (A0–A3)).
P	The reg field of the ModR/M byte selects a packed quadword MMX™ technology register.

Q	A ModR/M byte follows the opcode and specifies the operand. The operand is either an MMX™ technology register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register,an index register, a scaling factor, and a displacement.
R	The mod field of the ModR/M byte may refer only to a general register (for example, MOV (0F20-0F24, 0F26)).
S	The reg field of the ModR/M byte selects a segment register (for example, MOV (8C,8E)).
T	The reg field of the ModR/M byte selects a test register (for example, MOV (0F24,0F26)).
V	The reg field of the ModR/M byte selects a packed SIMD floating-point register.
W	An ModR/M byte follows the opcode and specifies the operand. The operand is either a SIMD floating-point register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register, an index register, a scaling factor, and a displacement
X	Memory addressed by the DS:SI register pair (for example, MOVS, CMPS, OUTS, or LODS).
Y	Memory addressed by the ES:DI register pair (for example, MOVS, CMPS, INS, STOS, or SCAS).

Operand Type Codes

a	Two one-word operands in memory or two double-word operands in memory, depending on operand-size attribute (used only by the BOUND instruction).
b	Byte, regardless of operand-size attribute.
c	Byte or word, depending on operand-size attribute.

d	Doubleword, regardless of operand-size attribute
dq	Double-quadword, regardless of operand-size attribute.
p	32-bit or 48-bit pointer, depending on operand-size attribute.
pi	Quadword MMX™ technology register (e.g. mm0)
ps	128-bit packed FP single-precision data.
q	Quadword, regardless of operand-size attribute.
s	6-byte pseudo-descriptor.
ss	Scalar element of a 128-bit packed FP single-precision data.
si	Doubleword integer register (e.g., eax)
v	Word or doubleword, depending on operand-size attribute.
w	Word, regardless of operand-size attribute.