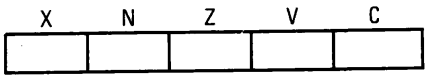
In the instruction set definitions, the CCR is shown as follows:



where:

X (extend)
Set to the value of the C bit for arithmetic operations. Otherwise not affected or set to a specified result.

N (negative)
Set if the most significant bit of the result is set. Cleared otherwise.

Z (zero)
Set if the result equals zero. Cleared otherwise.

V (overflow)
Set if arithmetic overflow occurs. This implies that the result cannot be represented in the operand size. Cleared otherwise.

C (carry)
Set if a carry out of the most significant bit of the operand occurs for an addition. Also set if a borrow occurs in a subtraction. Cleared otherwise.

4.2.1 Condition Code Computation

Most operations take a source operand and a destination operand, compute, and store the result in the destination location. Single-operand operations take a destination operand, compute, and store the result in the destination location. Table 4-10 lists each instruction and how it affects the condition code bits.

Table 4-10. Condition Code Computations (Sheet 1 of 2)

Operations	X	N	Z	V	C	Special Definition
ABCD	*	U	?	U	?	C = Decimal Carry $Z = Z \wedge \overline{Rm} \wedge \ldots \wedge \overline{R0}$
ADD, ADDI, ADDQ	*	*	*	?	?	$V = Sm \wedge Dm \wedge \overline{Rm} \vee \overline{Sm} \wedge \overline{Dm} \wedge Rm$ $C = Sm \wedge Dm \vee \overline{Rm} \wedge Dm \vee Sm \wedge \overline{Rm}$
ADDX	*	*	?	?	?	$V = Sm \wedge Dm \wedge \overline{Rm} \vee \overline{Sm} \wedge \overline{Dm} \wedge Rm$ $C = Sm \wedge Dm \vee \overline{Rm} \wedge Dm \vee Sm \wedge \overline{Rm}$ $Z = Z \wedge \overline{Rm} \wedge \ldots \wedge \overline{R0}$
AND, ANDI, EOR, EORI, MOVEQ, MOVE, OR, ORI, CLR, EXT, NOT, TAS, TST	—	*	*	0	0	
CHK	—	*	U	U	U	

Table 4-10. Condition Code Computations (Sheet 2 of 2)

Operations	X	N	Z	V	C	Special Definition
CHK2	—	U	?	U	?	Z = (R = LB) V (R = UB) C = (LB < = UB) Λ (IR < LB) V (R > UB)) V (UB < LB) Λ (R > UB) Λ (R < LB)
SUB, SUBI, SUBQ	*	*	*	?	?	V = $\overline{Sm}$ Λ $\underline{Dm}$ Λ $\overline{Rm}$ V $\underline{Sm}$ Λ $\overline{Dm}$ Λ Rm C = Sm Λ $\overline{Dm}$ V Rm Λ $\overline{Dm}$ V Sm Λ Rm
SUBX	*	*	?	?	?	V = $\overline{Sm}$ Λ $\underline{Dm}$ Λ $\overline{Rm}$ V $\underline{Sm}$ Λ $\overline{Dm}$ Λ Rm C = Sm Λ $\overline{Dm}$ V Rm Λ $\overline{Dm}$ V Sm Λ Rm Z = Z Λ $\overline{Rm}$ Λ . . . Λ $\overline{R0}$
CMP, CMPI, CMPM	—	*	*	?	?	V = $\overline{Sm}$ Λ $\underline{Dm}$ Λ $\overline{Rm}$ V $\underline{Sm}$ Λ $\overline{Dm}$ Λ Rm C = Sm Λ $\overline{Dm}$ V Rm Λ $\overline{Dm}$ V Sm Λ Rm
DIVS, DUVI	—	*	*	?	0	V = Division Overflow
MULS, MULU	—	*	*	?	0	V = Multiplication Overflow
SBCD, NBCD	*	U	?	U	?	C = Decimal Borrow Z = Z Λ $\overline{Rm}$ Λ . . . Λ $\overline{Ro}$
NEG	*	*	*	?	?	V = Dm Λ Rm C = Dm V Rm
NEGX	*	*	?	?	?	V = Dm Λ Rm C = Dm V Rm Z = Z Λ $\overline{Rm}$ Λ . . . Λ $\overline{R0}$
BTST, BCHG, BSET, BCLR	—	—	?	—	—	Z = $\overline{Dn}$
ASL	*	*	*	?	?	V = Dm Λ ($\overline{Dm-1}$ V . . . V $\overline{Dm-r}$) V $\overline{Dm}$ Λ (DM−1 V . . . +Dm−r) C = $\overline{Dm-r+1}$
ASL (R=0)	—	*	*	0	0	
LSL, ROXL	*	*	*	0	?	C = Dm−r+1
LSR (r=0)	—	*	*	0	0	
ROXL (r=0)	—	*	*	0	?	C=X
ROL	—	*	*	0	?	C=Dm−r+1
ROL (r=0)	—	*	*	0	0	
ASR, LSR, ROXR	*	*	*	0	?	C=Dr−1
ASR, LSR (r=0)	—	*	*	0	0	
ROXR (r=0)	—	*	*	0	?	C=X
ROR	—	*	*	0	?	C = Dr−1
ROR (r=0)	—	*	*	0	0	

— = Not Affected
U = Undefined, Result Meaningless
? = Other — See Special Definition
* = General Case
X = C
N = $\underline{Rm}$
Z = $\overline{Rm}$ Λ . . . Λ $\overline{R0}$
Sm = Source Operand — Most Significant Bit
Dm = Destination Operand — Most Significant Bit
Rm = Result Operand — Most Significant Bit
R = Register Tested
n = Bit Number
r = Shift Count
LB = Lower Bound
UB = Upper Bound
Λ = Boolean AND
V = Boolean OR
$\overline{Rm}$ = NOT Rm

4.2.2 Conditional Tests

Table 4-11 lists the condition names, encodings, and tests for the conditional branch and set instructions. The test associated with each condition is a logical formula using the current states of the condition codes. If this formula evaluates to one, the condition is true. If the formula evaluates to zero, the condition is false. For example, the T condition is always true, and the EQ condition is true only if the Z bit condition code is currently true.

Table 4-11. Conditional Tests

Mnemonic	Condition	Encoding	Test
T*	True	0000	1
F*	False	0001	0
HI	High	0010	$\overline{C}\cdot\overline{Z}$
LS	Low or Same	0011	$C+Z$
CC(HS)	Carry Clear	0100	$\overline{C}$
CS(LO)	Carry Set	0101	C
NE	Not Equal	0110	$\overline{Z}$
EQ	Equal	0111	Z
VC	Overflow Clear	1000	$\overline{V}$
VS	Overflow Set	1001	V
PL	Plus	1010	$\overline{N}$
MI	Minus	1011	N
GE	Greater or Equal	1100	$N\cdot V+\overline{N}\cdot\overline{V}$
LT	Less Than	1101	$N\cdot\overline{V}+\overline{N}\cdot V$
GT	Greater Than	1110	$N\cdot V\cdot\overline{Z}+\overline{N}\cdot\overline{V}\cdot\overline{Z}$
LE	Less or Equal	1111	$Z+N\cdot\overline{V}+\overline{N}\cdot V$

• = Boolean AND
+ = Boolean OR
$\overline{N}$ = Boolean NOT N

*Not available for the Bcc instruction.

4.3 INSTRUCTION SET SUMMARY

Table 4-12 provides an alphabetized listing of the M68000 instruction set listed by opcode, operation, and syntax. In the syntax descriptions, the left operand is the source operand, and the right operand is the destination operand. The following list contains the notations used in Table 4-12.

ADDX

Add Extended
(M68000 Family)

Operation: Source + Destination + X ♦ Destination

Assembler Syntax: ADDX Dy,Dx
ADDX –(Ay),–(Ax)

Attributes: Size = (Byte, Word, Long)

Description: Adds the source operand and the extend bit to the destination operand and stores the result in the destination location. The operands can be addressed in two different ways:

1. Data register to data register—The data registers specified in the instruction contain the operands.
2. Memory to memory—The address registers specified in the instruction address the operands using the predecrement addressing mode.

The size of the operation can be specified as byte, word, or long.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

X—Set the same as the carry bit.
N—Set if the result is negative; cleared otherwise.
Z—Cleared if the result is nonzero; unchanged otherwise.
V—Set if an overflow occurs; cleared otherwise.
C—Set if a carry is generated; cleared otherwise.

NOTE

Normally, the Z condition code bit is set via programming before the start of an operation. This allows successful tests for zero results upon completion of multiple-precision operations.

Subtract with Extend
(M68000 Family)

SUBX

Operation: Destination – Source – X ♦ Destination

Assembler Syntax: SUBX Dx,Dy
SUBX –(Ax),–(Ay)

Attributes: Size = (Byte, Word, Long)

Description: Subtracts the source operand and the extend bit from the destination operand and stores the result in the destination location. The instruction has two modes:

1. Data register to data register—the data registers specified in the instruction contain the operands.
2. Memory to memory—the address registers specified in the instruction access the operands from memory using the predecrement addressing mode.

The size of the operand is specified as byte, word, or long.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

X—Set to the value of the carry bit.
N—Set if the result is negative; cleared otherwise.
Z—Cleared if the result is nonzero; unchanged otherwise.
V—Set if an overflow occurs; cleared otherwise.
C—Set if a borrow occurs; cleared otherwise.

NOTE

Normally the Z condition code bit is set via programming before the start of an operation. This allows successful tests for zero results upon completion of multiple-precision operations.

Compare
(M68000 Family)

Operation: Destination – Source ♦ cc

Assembler Syntax: CMP <ea>, Dn

Attributes: Size = (Byte, Word, Long)

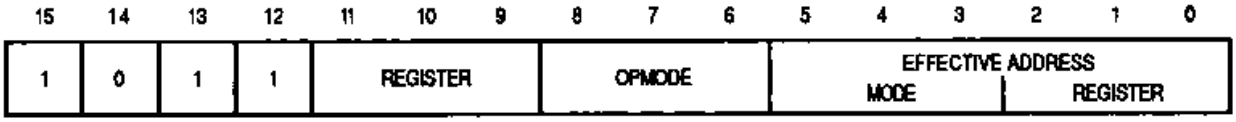
Description: Subtracts the source operand from the destination data register and sets the condition codes according to the result; the data register is not changed. The size of the operation can be byte, word, or long.

Condition Codes:

X	N	Z	V	C
—	*	*	*	*

X—Not affected.
N—Set if the result is negative; cleared otherwise.
Z—Set if the result is zero; cleared otherwise.
V—Set if an overflow occurs; cleared otherwise.
C—Set if a borrow occurs; cleared otherwise.

Instruction Format:



Instruction Fields:

Register field—Specifies the destination data register.

Opmode field

Byte	Word	Long	Operation
000	001	010	Dn – <ea>