

ADD

Add Binary

ADD

Operation: (Source)+(Destination)→Destination

Assembler Syntax ADD <ea>,Dn
ADD Dn,<ea>

Attributes: Size=(Byte,Word,Long)

Description: Add the source operand to the destination operand, and store the result in the destination location. The size of the operation may be specified to be byte, word, or long. The mode of the instruction indicates which operand is the source and which is the destination as well as the operand size.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

- N Set if the result is negative. Cleared otherwise.
- Z Set if the result is zero. Cleared otherwise.
- V Set if an overflow is generated. Cleared otherwise.
- C Set if a carry is generated. Cleared otherwise.
- X Set the same as the carry bit.

Instruction Format:

15 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0

1	1	0	1	Register	Op-Mode	Effective Address Mode	Register
---	---	---	---	----------	---------	------------------------	----------

Instruction Fields:

Register field — Specifies any of the eight data registers.

Op-Mode field —

Byte	Word	Long	Operation
000	001	010	(<Dn>)+(<ea>)→<Dn>
100	101	110	(<ea>)+(<Dn>)→<ea>

Effective Address field — Determines addressing mode:

- a. If the location specified is a source operand, then all addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	register number	d(An,Xi)	110	register number
An*	001	register number	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	111	010
-(An)	100	register number	d(PC,Xi)	111	011
d(An)	101	register number	Imm	111	100

* Word and Long only.

- b. If the location specified is a destination operand, then only alterable memory addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	—	—	d(An,Xi)	110	register number
An	—	—	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	—	—
-(An)	100	register number	d(PC,Xi)	—	—
d(An)	101	register number	Imm	—	—

- Notes:**
1. If the destination is a data register, then it cannot be specified by using the destination <ea> mode, but must use the destination Dn mode instead.
 2. ADDA is used when the destination is an address register. ADDI and ADDQ are used when the source is immediate data. Most assemblers automatically make this decision.

MOVE

Move Data From Source to Destination

MOVE

Operation: (Source)→Destination

Assembler Syntax MOVE <ea>,<ea>

Attributes: Size=(Byte,Word,Long)

Description: Move the content of the source to the destination location. The data is examined as it is moved, and the condition codes set accordingly. The size of the operation may be specified to be byte, word or long.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

N Set if the result is negative. Cleared otherwise.

Z Set if the result is zero. Cleared otherwise.

V Always cleared.

C Always cleared.

X Not affected.

Instruction Format:

15 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0

0	0	Size	Destination Register	Mode	Source Mode	Register
---	---	------	----------------------	------	-------------	----------

Instruction Fields:

Size field — Specifies the size of the operand to be moved.

01 — byte operation.

11 — word operation.

10 — long operation.

Destination Effective Address field — Specifies the destination location. Only data alterable addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	register number	d(An,Xi)	110	register number
An	—	—	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	—	—
-(An)	100	register number	d(PC,Xi)	—	—
d(An)	101	register number	Imm	—	—

Source Effective Address field — Specifies the source operand. All addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	register number	d(An,Xi)	110	register number
An*	001	register number	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	111	010
-(An)	100	register number	d(PC,Xi)	111	011
d(An)	101	register number	Imm	111	100

* For byte size operation, address register direct is not allowed.

- Notes:**
1. MOVEA is used when the destination is an address register. Most assemblers automatically make this distinction.
 2. MOVEQ can also be used for certain operations on data registers.

ADDI

Add Immediate

ADDI

Operation: Immediate Data + (Destination)→Destination

Assembler Syntax ADD #<data>,<ea>

Attributes: Size=(Byte,Word,Long)

Description: Add the immediate data to the destination operand, and store the result in the destination location. The size of the operation may be specified to be byte, word, or long. The size of the immediate data matches the operation size.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

- N Set if the result is negative. Cleared otherwise.
- Z Set if the result is zero. Cleared otherwise.
- V Set if an overflow is generated. Cleared otherwise.
- C Set if a carry is generated. Cleared otherwise.
- X Set the same as the carry bit.

Instruction Format:

15 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0

0	0	0	0	0	1	1	0	Size	Effective Address	
									Mode	Register
Word Data (16 bits)								Byte Data (8 bits)		
Long Data (32 bits, including previous word)										

Instruction Fields:

Size field — Specifies the size of the operation.

- 00 — byte operation.
- 01 — word operation.
- 10 — long operation.

Effective Address field — Specifies the destination operand. Only data alterable addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	register number	d(An,Xi)	110	register number
An	—	—	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	—	—
-(An)	100	register number	d(PC,Xi)	—	—
d(An)	101	register number	Imm	—	—

Immediate field — (Data immediately following the instruction):

- If size=00, then the data is the low order byte of the immediate word.
- If size=01, then the data is the entire immediate word.
- If size=10, then the data is the next two immediate words.

SUB

Subtract Binary

SUB

Operation: (Destination)-(Source)→Destination

Assembler Syntax
SUB <ea>,Dn
SUB Dn,<ea>

Attributes: Size=(Byte,Word,Long)

Description: Subtract the source operand from the destination operand, and store the result in the destination. The size of the operation may be specified to be byte, word, or long. The mode of the instruction indicates which operand is the source and which is the destination as well as the operand size.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

- N Set if the result is negative. Cleared otherwise.
- Z Set if the result is zero. Cleared otherwise.
- V Set if an overflow is generated. Cleared otherwise.
- C Set if a carry is generated. Cleared otherwise.
- X Set the same as the carry bit.

Instruction Format:

15 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0

1	0	0	1	Register	Op-Mode	Effective Address	
						Mode	Register

Instruction Fields:

Register field — Specifies any of the eight data registers.

Op-Mode field —

Byte	Word	Long	Operation
000	001	010	(<Dn>)-(<ea>)→<Dn>
100	101	110	(<ea>)-(<Dn>)→<ea>

Effective Address field — Determines addressing mode:

- a. If the location specified is a source operand, then all addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	000	register number	d(An,Xi)	110	register number
An*	001	register number	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	111	010
-(An)	100	register number	d(PC,Xi)	111	011
d(An)	101	register number	Imm	111	100

* For byte size operation, address register direct is not allowed

- b. If the location specified is a destination operand, then only alterable memory addressing modes are allowed as shown:

Addressing Mode	Mode	Register	Addressing Mode	Mode	Register
Dn	—	—	d(An,Xi)	110	register number
An	—	—	Abs.W	111	000
(An)	010	register number	Abs.L	111	001
(An)+	011	register number	d(PC)	—	—
-(An)	100	register number	d(PC,Xi)	—	—
d(An)	101	register number	Imm	—	—

- Notes:**
1. If the destination is a data register, then it cannot be specified by using the destination <ea> mode, but must use the destination Dn mode instead.
 2. SUBA is used when the destination is an address register. SUBI and SUBQ are used when the source is immediate data. Most assemblers automatically make this distinction.