Carga de constante: MOVEQ

Integer Instructions

MOVEQ — Move Quick — MOVEQ

Operation: Immediate Data → Destination

Assembler Syntax: MOVEQ # < data > ,Dn

Attributes: Size = Long

Description: Moves a byte of immediate data to a 32-bit data register. The data in an 8-bit field within the operation word is sign- extended to a long operand in the data register as it is transferred.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

X — not affected
N — set if the result is negative; cleared otherwise
Z — set if the result is zero; cleared otherwise
V — always cleared
C — always cleared

Instruction Format:

15	14	13	12	11 10 9	8	7 6 5 4 3 2 1 0
0	1	1	1	REGISTER	0	DATA

FORMATO DA INSTRUÇÃO: 15 - 12 : OPCODE
11 - 9 : N° DO REGISTRADOR
8 - 0 : DADO

Cópia de Valor Entre Registradores:MOVE/MOVEA - Move dados da origem para o destino

MOVE MOVEA — Move Data from Source to Destination — MOVE MOVEA

Operation: Source → Destination

Assembler Syntax:
MOVE < ea > , < ea >
MOVEA <ea>, An

Attributes: Size = Byte, Word, Long

Description: Moves the data at the source to the destination location and sets the condition codes according to the data. The size of the operation may be specified as byte, word, or long word.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

X — not affected
N — set if the result is negative; cleared otherwise
Z — set if the result is zero; cleared otherwise
V — always cleared
C — always cleared

Instruction Format:

15	14	13 12	11 10 9	8 7 6	5 4 3	2 1 0
0	0	SIZE	DESTINATION REGISTER	DESTINATION MODE	SOURCE MODE	SOURCE REGISTER

FORMATO DA INSTRUÇÃO: 15 - 12 : OPCODE
11 - 6 : N° DO REGISTRADOR FONTE
5 - 0 : N° DO REGISTRADOR DESTINO

Soma de dois valores: ADDX- Adição

Integer Instructions

ADDX **Add Extended** **ADDX**

Operation: Source + Destination + X → Destination

Assembler ADDX Dy,Dx

Attributes: Size = 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 from data register to data register—where the data registers specified in the instruction contain the operands.

The size of the operation is specified as a long word.

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

Instruction Format:

15	14	13	12	11 10 9	8	7	6	5	4	3	2 1 0
1	1	0	1	REGISTER Rx	1	1	0	0	0	0	REGISTER Ry

FORMATO DA INSTRUÇÃO: 15 - 12 : OPCODE

11 - 9 : N° DO REGISTRADOR DESTINO

8 - 0 : N° DO REGISTRADOR FONTE

Subtração de dois valores: SUBX- Subtração

Integer Instructions

SUBX — Subtract with Extend — SUBX

Operation: Destination – Source – X → Destination

Assembler Syntax: SUBX Dx,Dy

Attributes: Size = Long

Description: Subtracts the source operand and the extend bit from the destination operand and stores the result in the destination.

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

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	Dy			1	1	0	0	0	0	Dx		

FORMATO DA INSTRUÇÃO: 15 - 12 : OPCODE
11 - 9 : N° DO REGISTRADOR DESTINO
8 - 0 : N° DO REGISTRADOR FONTE

Desvio incondicional: JUMP - Salto incondicional

JMP **Jump** **JMP**

Operation: Destination Address → PC

Assembler Syntax: JMP < ea >

Attributes: Unsized

Description: Program execution continues at the effective address specified by the instruction. The addressing mode for the effective address must be a control addressing mode.

Condition Codes:

Not affected

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	1	1	EFFECTIVE ADDRESS MODE			REGISTER		

FORMATO DA INSTRUÇÃO: 15 - 12 : OPCODE

11 - 0 : N° DO REGISTRADOR COM O ENDEREÇO DE DESTINO DO SALTO

Especificação da Codificação das Instruções

OPCODES

0000 - NOP
0010 - MOVEQ
0011 - MOVE
0100 - ADD
0110 - SUBX
1111 - JUMP

SELEÇÃO DE OPERAÇÃO DA ULA

00 - SOMA
01 - SUBTRAÇÃO
11 - ENT0

CÓDIGO ASSEMBLY

instrução assembly	código da instrução
1 MOVEQ # 5, D3	- 0010 011 000000101
2 MOVEQ # 8, D4	- 0010 100 000001000
3 MOVE D3, D5	- 0011 101 000000011
ADDX D4, D5	- 0100 101 000000100
4 MOVEQ 1, D1	- 0010 001 000000001
SUBX D1, D5	- 0110 101 000000001
5 MOVEQ # 20, D2	- 0010 010 000010100
JUMP D2	- 1111 000000000 010
6 MOVE D5, D3	- 0011 011 000000101
7 MOVEQ # 2, D6	- 0010 110 000000010
JUMP D6	- 1111 000000000 110