

Motorola 68000 (simplificado)

Nesta versão do Motorola 68000, tanto instruções quanto dados têm o mesmo tamanho de palavra, 16-bit.

O número de instruções foram reduzidos em relação à versão original, pois se trata de um processador CISC com mais de um CORE para cálculos em paralelo.

Tanto a sintaxe assembler quanto a configuração das instruções em código de máquina estão listadas após na seção “3 – Tipos de instrução”.

Os tipos de endereçamento também foram simplificados para facilitar o desenvolvimento deste simulador.

1 - Tipos de endereçamento

Os tipos de endereçamento são:

$n \in \{0, 1, \dots, 7\}$
 $\langle ea \rangle \in \{0, 1, \dots, 2^{16} - 1\}$
 $x \in \{0, 1, \dots, 2^{16} - 1\}$

Registrador de dados direto	Dn
Registrador de endereços direto	An
Registrador de endereços indireto	(An)
Endereço direto	$\langle ea \rangle$
Imediato	#x

2 – Campos de instrução

Os campos de instrução são os seguintes:

[15	10] [9	6] [5	3] [2	0]
Tipo instrução	Tipo end	Reg origem	Reg destino	

“Tipo de instrução” utiliza 6-bits para representar qual operação o processador deve executar. O conteúdo deste campo está melhor especificado após esta seção.

“Tipo end” utiliza 4-bits para representar alguns dos tipos de endereçamento possíveis. São esses:

$$[Dn, An, (An), <ea>] \times [Dn, An, (An), <ea>]$$

“Tipo de end. à esquerda” 00 – 00 “Tipo de end. à direita” .

Exemplos:

Dn, Dn	[0000]
Dn, An	[0001]
An, (An)	[0110]
An, <ea>	[0111]
(An), Dn	[1000]

Totalizando assim 16 combinações. O endereçamento imediato é informado pelo campo “Tipo de instrução”.

3 – Tipos de instrução

Op1 ∈ {Dn, An, (An), <ea>}

Op2 ∈ {Dn, An, (An), <ea>}

<dado> ∈ {0, 1, ..., $2^{16} - 1$ }

Rn ∈ {D0, D1, ..., D7, A0, A1, ..., A7}

Sintaxe assembler

Código de máquina					<- Condição de operação
X	N	Z	V	C	

X = Extended

N = Negative

Z = Zero

V = Overflow

C = Carry

ADD Op1 Op2

000000					----	---	---
*	*	*	*	*			

ADDI #<dado> Op2

000001					XX--	XXX	---
*	*	*	*	*			

AND Op1 Op2

000010					----	---	---
x	*	*	0	0			

ANDI #<dado> Op2

000011					XX--	XXX	---
x	*	*	0	0			

CLR Op1

000100					--XX	---	XXX
x	0	1	0	0			

CMP Op1 Op2

000101					----	---	---
x	*	*	*	*			

DIVS Op1 Op2

000110					----	---	---
x	*	*	*	0			

EOR Op1 Op2

000111					----	---	---
x	*	*	0	0			

EORI #<data> Op2

001000					xx--	xxx	---
x	*	*	0	0			

EXG Rn, Rn

001001					x-x-	---	---
x	x	x	x	x			

JMP <ea>

001010					xxxx	xxx	xxx
x	x	x	x	x			

LSL #<data> Op2

001011					xx--	xxx	---
*	*	*	0	*			

LSR #<data> Op2

001100					xx--	xxx	---
*	*	*	0	*			

MOVE Op1 Op2

001101					----	---	---
x	*	*	0	0			

MULS Op1 Op2

001110					----	---	---
x	*	*	0	0			

NEG Op1

001111					--XX	---	xxx
*	*	*	*	*			

NOP

010000					xxxx	xxx	xxx
x	x	x	x	x			

NOT Op1

010001					--XX	---	xxx
x	*	*	0	0			

OR Op1 Op2

010010					----	---	---
x	*	*	0	0			

ORI #<data> Op2

010011					XX--	xxx	---
x	*	*	0	0			

SUB Op1 Op2

010100					----	---	---
*	*	*	*	*			

SUBI #<data> Op2

010101					XX--	xxx	---
*	*	*	*	*			

BEQ Op1

010110					--XX	---	xxx
x	x	x	x	x			

BGE Op1

010111					--XX	---	xxx
x	x	x	x	x			

BGT Op1

011000					--XX	---	xxx
x	x	x	x	x			

BLE Op1

011001					--XX	---	xxx
x	x	x	x	x			

BLT Op1

011010					--XX	---	xxx
x	x	x	x	x			

BMI Op1

011011					--XX	---	xxx
x	x	x	x	x			

BPL Op1

011100					--XX	---	xxx
x	x	x	x	x			

HALT

011101					xxxx	xxx	xxx
x	x	x	x	x			