

mnemônico		codigo binário significado	
init ax	0	0000 0000	inicializa valor de ax com zero -> ax = 0
move ax,bx	1	0000 0001	ax = bx
move ax,cx	2	0000 0010	ax = cx
move bx,ax	3	0000 0011	bx = ax
move cx,ax	4	0000 0100	cx = ax
move ax,[	5	0000 0101	move conteudo da posicao X da memoria p/ax. Ex: move ax,[50] significa: move conteúdo da posição 50 para AX
move ax,[bx+	6	0000 0110	move conteudo da posicao [bx+X] da memoria p/ax.Ex: move ax,[bx+2]
move ax,[bp-	7	0000 0111	move conteudo da posicao [bp-X] para AX. Ex: move ax,[bp-2]
move ax,[bp+	8	0000 1000	move conteudo da posicao [bp+X] para AX. Ex: move ax,[bp+2]
move [	9	0000 1001	move valor de ax para a posicao X de memoria. Ex: move [30],ax -> significa move o valor de AX para a posicao 30 da memória
move [bx+	10	0000 1010	move valor de ax para a posicao de memoria indexada por [bx+valor]. Ex: move [bx+3],ax -> significa move o valor de AX para a posição [bx+3];se bx=50, o resultado é: move [53],ax
move bp,sp	11	0000 1011	bp = sp
move sp,bp	12	0000 1100	sp = bp
add ax,bx	13	0000 1101	ax = ax + bx
add ax,cx	14	0000 1110	ax = ax + cx
add bx,cx	15	0000 1111	bx = bx + cx
sub ax,bx	16	0001 0000	ax = ax - bx
sub ax,cx	17	0001 0001	ax = ax -cx
sub bx,cx	18	0001 0010	bx = bx-cx
inc ax	19	0001 0011	ax++
inc bx	20	0001 0100	bx++
inc cx	21	0001 0101	cx++
dec ax	22	0001 0110	ax--
dec bx	23	0001 0111	bx--
dec cx	24	0001 1000	cx--
test ax0,	25	0001 1001	if ax==0. Ex: testAX0, 150. Ex: se AX==0 entao IP = 150
jmp	26	0001 1010	ip = posicao de memoria. Ex: jmp 150 ao executar ip = 150, caso contrario não faz nada
call	27	0001 1011	empilha ip e faz ip = posicao de memoria, ou seja mem[sp] = ip; sp--; ip=endereço. Ex.: call 150 -> se ip = 100, ao executar a instrucao, empilha o valor 101 (endereço da proxima instrucao) e faz ip= 150
ret	28	0001 1100	sp++; ip = mem[sp] ->topo da pilha. Ex: ret - se ao executar esta instrucao o topo da pilha tiver o valor de 101 entao ip = 101
in ax	29	0001 1101	ax = valor do teclado. Simula uma tecla pressionada
out ax	30	0001 1110	placa de video = ax. Simula uma operacao de escrita
push ax	31	0001 1111	mem[sp] = ax; sp = sp-1
push bx	32	0010 0000	mem[sp] = bx; sp = sp-1
push cx	33	0010 0001	mem[sp] = cx; sp = sp-1
push bp	34	0010 0010	mem[sp] = bp; sp = sp-1
pop bp	35	0010 0011	sp = sp+1; bp =mem[sp];
pop cx	36	0010 0100	sp = sp+1;cx =mem[sp];
pop bx	37	0010 0101	sp = sp+1; bx =mem[sp];
pop ax	38	0010 0110	sp = sp+1; ax =mem[sp];
nop	39	0010 0111	no operation
halt	40	0010 1000	pára o processador
dec sp	41	0010 1001	sp = sp -1
move [bp-	42	0010 1010	move conteudo de AX para a posicao [bp-X]. Ex: move [bp-2],ax
move [bp+	43	0010 1011	move conteudo de AX para a posicao [bp+X]. Ex: move [bp+2],ax
move ax,{	44	0010 1100	move valor imediato entre chaves para o registrador ax. Ex. move ax,{50} move para ax o valor 50
test axEqbx,	45	0010 1101	if ax==bx. Ex: testAXEqBX, 150. Ex: se AX==BX entao IP = 150
inc sp	46	0010 1110	sp = sp + 1
move ax,sp	47	0010 1111	ax = sp
move sp,ax	48	0011 0000	sp = ax
move ax,bp	49	0011 0001	ax = bp
move bp,ax	50	0011 0010	bp =ax
iret	51	0011 0011	retorna de uma interrupcao. Equivale a: pop cx,pop bx,pop ax,pop bp, ret
int	52	0011 0100	gera uma interrupcao. Equivale a: push ip,push bp, push ax,push bx, push cx, ip=mem[ numeroInt]. Ex: int 3 -> ip=mem[3]

a29	53	0011 0101	
a30	54	0011 0110	
a31	55	0011 0111	
a32	56	0011 1000	
a33	57	0011 1001	
a34	58	0011 1010	
a35	59	0011 1011	
a36	60	0011 1100	
a37	61	0011 1101	
a38	62	0011 1110	
a39	63	0011 1111	
a40	64	0100 0000	
a41	65	0100 0001	
a42	66	0100 0010	
a43	67	0100 0011	
a44	68	0100 0100	
a45	69	0100 0101	
a46	70	0100 0110	
a47	71	0100 0111	
a48	72	0100 1000	
a49	73	0100 1001	
a50	74	0100 1010	
a51	75	0100 1011	
a52	76	0100 1100	
a53	77	0100 1101	
a54	78	0100 1110	
a55	79	0100 1111	
a56	80	0101 0000	
a57	81	0101 0001	
a58	82	0101 0010	
a59	83	0101 0011	
a60	84	0101 0100	
a61	85	0101 0101	
a62	86	0101 0110	
a63	87	0101 0111	
a64	88	0101 1000	
a65	89	0101 1001	
a66	90	0101 1010	
a67	91	0101 1011	
a68	92	0101 1100	
a69	93	0101 1101	
a70	94	0101 1110	
a71	95	0101 1111	
a72	96	0110 0000	
a73	97	0110 0001	
a74	98	0110 0010	
a75	99	0110 0011	
a76	100	0110 0100	
a77	101	0110 0101	
a78	102	0110 0110	
a79	103	0110 0111	
a80	104	0110 1000	
a81	105	0110 1001	
a82	106	0110 1010	
a83	107	0110 1011	
a84	108	0110 1100	
a85	109	0110 1101	
a86	110	0110 1110	
a87	111	0110 1111	
a88	112	0111 0000	
a89	113	0111 0001	
a90	114	0111 0010	
a91	115	0111 0011	
a92	116	0111 0100	
a93	117	0111 0101	
a94	118	0111 0110	
a95	119	0111 0111	
a96	120	0111 1000	
a97	121	0111 1001	
a98	122	0111 1010	
a99	123	0111 1011	
a100	124	0111 1100	
a101	125	0111 1101	

a102	126	0111 1110	
a103	127	0111 1111	
a104	128	1000 0000	
a105	129	1000 0001	
a106	130	1000 0010	
a107	131	1000 0011	
a108	132	1000 0100	
a109	133	1000 0101	
a110	134	1000 0110	
a111	135	1000 0111	
a112	136	1000 1000	
a113	137	1000 1001	
a114	138	1000 1010	
a115	139	1000 1011	
a116	140	1000 1100	
a117	141	1000 1101	
a118	142	1000 1110	
a119	143	1000 1111	
a120	144	1001 0000	
a121	145	1001 0001	
a122	146	1001 0010	
a123	147	1001 0011	
a124	148	1001 0100	
a125	149	1001 0101	
a126	150	1001 0110	
a127	151	1001 0111	
a128	152	1001 1000	
a129	153	1001 1001	
a130	154	1001 1010	
a131	155	1001 1011	
a132	156	1001 1100	
a133	157	1001 1101	
a134	158	1001 1110	
a135	159	1001 1111	
a136	160	1010 0000	
a137	161	1010 0001	
a138	162	1010 0010	
a139	163	1010 0011	
a140	164	1010 0100	
a141	165	1010 0101	
a142	166	1010 0110	
a143	167	1010 0111	
a144	168	1010 1000	
a145	169	1010 1001	
a146	170	1010 1010	
a147	171	1010 1011	
a148	172	1010 1100	
a149	173	1010 1101	
a150	174	1010 1110	
a151	175	1010 1111	
a152	176	1011 0000	
a153	177	1011 0001	
a154	178	1011 0010	
a155	179	1011 0011	
a156	180	1011 0100	
a157	181	1011 0101	
a158	182	1011 0110	
a159	183	1011 0111	
a160	184	1011 1000	
a161	185	1011 1001	
a162	186	1011 1010	
a163	187	1011 1011	
a164	188	1011 1100	
a165	189	1011 1101	
a166	190	1011 1110	
a167	191	1011 1111	
a168	192	1100 0000	
a169	193	1100 0001	
a170	194	1100 0010	
a171	195	1100 0011	
a172	196	1100 0100	
a173	197	1100 0101	
a174	198	1100 0110	

<b>a175</b>	199	<b>1100 0111</b>	
<b>a176</b>	200	<b>1100 1000</b>	
<b>a177</b>	201	<b>1100 1001</b>	
<b>a178</b>	202	<b>1100 1010</b>	
<b>a179</b>	203	<b>1100 1011</b>	
<b>a180</b>	204	<b>1100 1100</b>	
<b>a181</b>	205	<b>1100 1101</b>	
<b>a182</b>	206	<b>1100 1110</b>	
<b>a183</b>	207	<b>1100 1111</b>	
<b>a184</b>	208	<b>1101 0000</b>	
<b>a185</b>	209	<b>1101 0001</b>	
<b>a186</b>	210	<b>1101 0010</b>	
<b>a187</b>	211	<b>1101 0011</b>	
<b>a188</b>	212	<b>1101 0100</b>	
<b>a189</b>	213	<b>1101 0101</b>	
<b>a190</b>	214	<b>1101 0110</b>	
<b>a191</b>	215	<b>1101 0111</b>	
<b>a192</b>	216	<b>1101 1000</b>	
<b>a193</b>	217	<b>1101 1001</b>	
<b>a194</b>	218	<b>1101 1010</b>	
<b>a195</b>	219	<b>1101 1011</b>	
<b>a196</b>	220	<b>1101 1100</b>	
<b>a197</b>	221	<b>1101 1101</b>	
<b>a198</b>	222	<b>1101 1110</b>	
<b>a199</b>	223	<b>1101 1111</b>	
<b>a200</b>	224	<b>1110 0000</b>	
<b>a201</b>	225	<b>1110 0001</b>	
<b>a202</b>	226	<b>1110 0010</b>	
<b>a203</b>	227	<b>1110 0011</b>	
<b>a204</b>	228	<b>1110 0100</b>	
<b>a205</b>	229	<b>1110 0101</b>	
<b>a206</b>	230	<b>1110 0110</b>	
<b>a207</b>	231	<b>1110 0111</b>	
<b>a208</b>	232	<b>1110 1000</b>	
<b>a209</b>	233	<b>1110 1001</b>	
<b>a210</b>	234	<b>1110 1010</b>	
<b>a211</b>	235	<b>1110 1011</b>	
<b>a212</b>	236	<b>1110 1100</b>	
<b>a213</b>	237	<b>1110 1101</b>	
<b>a214</b>	238	<b>1110 1110</b>	
<b>a215</b>	239	<b>1110 1111</b>	
<b>a216</b>	240	<b>1111 0000</b>	
<b>a217</b>	241	<b>1111 0001</b>	
<b>a218</b>	242	<b>1111 0010</b>	
<b>a219</b>	243	<b>1111 0011</b>	
<b>a220</b>	244	<b>1111 0100</b>	
<b>a221</b>	245	<b>1111 0101</b>	
<b>a222</b>	246	<b>1111 0110</b>	
<b>a223</b>	247	<b>1111 0111</b>	
<b>a224</b>	248	<b>1111 1000</b>	
<b>a225</b>	249	<b>1111 1001</b>	
<b>a226</b>	250	<b>1111 1010</b>	
<b>a227</b>	251	<b>1111 1011</b>	
<b>a228</b>	252	<b>1111 1100</b>	
<b>a229</b>	253	<b>1111 1101</b>	
<b>bp+</b>	254	<b>1111 1110</b>	
<b>bp-</b>	255	<b>1111 1111</b>	