

	Codificação das instruções					
	operação	opcode	formato	Operação		TAMANHO
	ADD	0001	opcode + reg + xxxxxxxx	$A \leftarrow A + \text{reg}$	instruções	16 bits
	ADDI	0010	opcode + xxxx + cte	$A \leftarrow A + \text{cte}$	opcode	4 bits
	SUB	0011	opcode + reg + xxxxxxxx	$A \leftarrow A - \text{reg}$	endereço dos regs.	3 bits
	SUBI	0100	opcode + xxxx + cte	$A \leftarrow A - \text{cte}$	endereço de memória	7 bits
	CMP	1000	opcode + reg + xxxxxxxx	$A - \text{reg}$	constante	8 bits
	JUMP	1111	opcode + xxxxx + end	$PC \leftarrow \text{end}$		
	JZ	1001	opcode + xxxxx + end_rel	if(Z=1) then $PC \leftarrow PC + \text{end_rel}$		
	JN	1010	opcode + xxxxx + end_rel	if(N=1) then $PC \leftarrow PC + \text{end_rel}$		
	JP	1100	opcode + xxxxx + end_rel	if(N=0 and Z=0) then $PC \leftarrow PC + \text{end_rel}$		
	MOV	0101	opcode + reg + d + xxxxxxxx	if(d=0) then ($A \leftarrow \text{reg}$) else ($\text{reg} \leftarrow A$)		
	LD	0110	opcode + reg + d + cte	if(d=0) then ($A \leftarrow \text{cte}$) else ($\text{reg} \leftarrow \text{cte}$)		
	SW	1101	opcode + reg + xxxxxxxx	$\text{RAM}(\text{reg}) \leftarrow A$		
	LW	1110	opcode + reg + xxxxxxxx	$A \leftarrow \text{RAM}(\text{reg})$		
	CJNE	1011	opcode + reg + xx + end	if(A!=reg) then $PC \leftarrow \text{end}$		
	Onde:					
	reg identifica o registrador fonte					
	cte é uma constante em complemento de 2					
	x são bits irrelevantes					
	end é um endereço de memória destino					
	end_rel é uma constante em complemento de 2 que determina o endereço de memória destino					
	Z é a flag ZERO					
	N é a flag NEGATIVE					
	d é um bit que decide se a instrução MOV copia do reg1 para o acumulador (0) ou do acumulador pro reg1 (1)					
	d é um bit que decide se o a instrução LD carrega a constante no reg1 (1) ou no acumulador (0)					
	As instruções de memória SW e LW usam reg como ponteiro para um endereço da RAM					