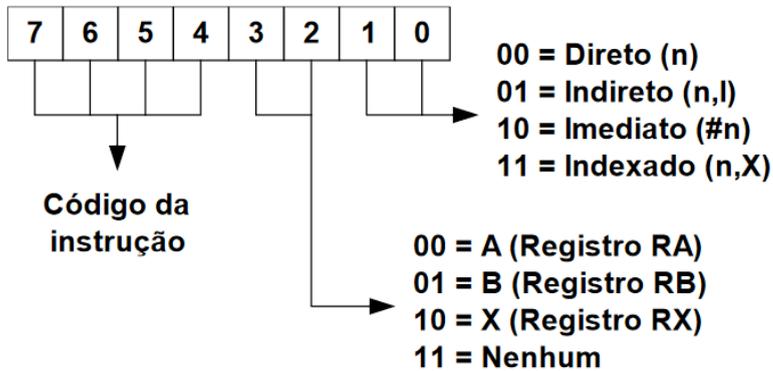




## Conjunto de instruções do RAMSES



H	Código e Operando	Endereço (2º byte)	Instrução	Operação	N	Z	C
0x	0000 xxxx	-	NOP	Nenhuma operação			
1?	0001 rmmm	dd8	STR rr dd8	MEM(dd8) ← rr			
2?	0010 rmmm	dd8	LDR rr dd8	rr ← MEM(dd8)	t	t	
3?	0011 rmmm	dd8	ADD rr dd8	rr ← rr + MEM(dd8)	t	t	t
4?	0100 rmmm	dd8	OR rr dd8	rr ← rr ∨ MEM(dd8)	t	t	
5?	0101 rmmm	dd8	AND rr dd8	rr ← rr ∧ MEM(dd8)	t	t	
6?	0110 rrx	-	NOT rr	rr ← ¬ rr	t	t	
7?	0111 rmmm	dd8	SUB rr dd8	rr ← rr - MEM(dd8)	t	t	t <sup>1</sup>
8?	1000 xxmm	dd8	JMP dd8	PC ← dd8 <sup>2</sup>			
9?	1001 xxmm	dd8	JN dd8	if N=1 then PC ← dd8 <sup>2</sup>			
A?	1010 xxmm	dd8	JZ dd8	if Z=1 then PC ← dd8 <sup>2</sup>			
B?	1011 xxmm	dd8	JC dd8	if C=1 then PC ← dd8 <sup>2</sup>			
C?	1100 xxmm	dd8	JSR dd8	MEM(dd8) ← PC PC ← dd8 + 1 <sup>2</sup>			
D?	1101 rrx	-	NEG rr	rr ← 0 - rr, ou seja, rr ← ¬ rr + 1	t	t	t
E?	1110 rrx	-	SHR rr	0 → [ registrador ] → C	t	t	t
Fx	1111 xxxx	-	HLT	Para a execução			

rr - indica um registrador (A, B ou X);

mm - indica um modo de endereçamento (direto, indireto, imediato ou indexado);

x - indica que o bit não importa para a execução da instrução;

dd8 - representa um endereço de 8 bits;

t - indica que o código de condição é testado pela unidade de controle e ajustado de acordo;

A ausência de t indica que o código de condição não é alterado e mantém seu estado anterior;

(1) o carry gerado na instrução SUB é o inverso do borrow, ou seja, C=1 indica que não houve borrow e C=0 indica que ocorreu borrow;

(2) as instruções de desvio no modo imediato são tratadas como NOP e o segundo byte é ignorado;

Características das sub-rotinas no RAMSES		
Recursividade	Não permitida	A capacidade de uma sub-rotina de chamar-se a si própria
Reentrância	Não permitida	A possibilidade de vários programas compartilharem uma única cópia da mesma sub-rotina
Aninhamento	Níveis ilimitados	A possibilidade de uma sub-rotina ser chamada dentro de outra sub-rotina

Passagem de parâmetros:

Quanto ao tipo dos parâmetros	
Passagem por VALOR	Passagem por REFERÊNCIA
<ul style="list-style-type: none"> <li>O parâmetro é o DADO propriamente dito</li> <li>Antes da chamada copia-se o VALOR original para a memória ou registrador</li> <li>Dentro da rotina utiliza-se a cópia da memória ou registrador (modo direto)</li> <li>Não permite a atualização do valor original, só é capaz de alterar o parâmetro local</li> </ul>	<ul style="list-style-type: none"> <li>O parâmetro é o PONTEIRO para o dado</li> <li>Antes da chamada – Copia-se o ENDEREÇO do valor original para a memória ou registrador – O ponteiro é passado por VALOR</li> <li>Dentro da rotina – Utiliza-se o acesso aos dados através do ponteiro – Modo INDIRETO do RAMSES</li> <li>Pode-se alterar o valor original que foi passado – Alteração através do ponteiro</li> <li>Não permite a alteração do ponteiro original (se existir)</li> </ul>
<p>LDR A vetor+3 ; Copia o valor "vetor[3]" para "param"                      STR A param                      JSR Inc                      HLT</p> <p>Inc: NOP                      LDR A param ; Operação realizada sobre a cópia                      ADD A #1                      STR A param ; Altera apenas o parâmetro copiado                      JMP Inc,l ; Não altera o valor original em "vetor[3]"</p>	<p>LDR A #vetor+3 ; Salva o endereço de "vetor[3]"                      STR A param                      JSR Inc                      HLT</p> <p>Inc: NOP                      LDR A param,l ; Operação utilizando ponteiro                      ADD A #1                      STR A param,l ; Altera o valor original no "vetor[3]"                      JMP Inc,l</p>
Quanto ao local de armazenamento dos parâmetros	
Passagem por REGISTRADOR	Passagem em MEMÓRIA
<ul style="list-style-type: none"> <li>O parâmetro está nos registradores do processador</li> <li>Antes da chamada copia-se para um REGISTRADOR o valor original ou ponteiro</li> <li>Dentro da rotina                             <ul style="list-style-type: none"> <li>Se for um VALOR, usa-se diretamente</li> <li>Se for um PONTEIRO, depende dos modos de endereçamento</li> </ul> </li> <li>É muito eficiente</li> </ul>	<ul style="list-style-type: none"> <li>Em área específica de dados                             <ul style="list-style-type: none"> <li>Ver exemplos "por valor" e "por referência"</li> </ul> </li> <li>Em área após a chamada                             <ul style="list-style-type: none"> <li>A "área específica" está imediatamente após a chamada da rotina</li> </ul> </li> <li>É menos eficiente do que a passagem por registrador</li> <li>A rotina deve efetuar o ajuste do endereço de retorno</li> </ul>
<p>LDR B vetor+3 ; Copia o "vetor[3]" para o RB                      LDR X #vetor ; Salva o endereço de "vetor[0]" em RX                      JSR Inc                      HLT</p> <p>Inc: NOP                      ADD B #1 ; Operação sobre o valor copiado (RB)</p> <p>LDR A #1                      ADD A 3,X ; Operação utilizando ponteiro (RX)                      STR A 3,X ; Altera o valor original no "vetor[3]"</p> <p>JMP Inc,l</p>	<p>JSR Soma ; Soma os 2 bytes que seguem                      DB 5 ; Parâmetro 1                      DB 7 ; Parâmetro 2                      HLT</p> <p>Soma: NOP                      LDR A Soma,l ; Pega o parâmetro "p1"</p> <p>LDR B Soma ; Incrementa o ponteiro para "p2"                      ADD B #1                      STR B Soma ; Ajusta o endereço de retorno</p> <p>ADD A Soma,l ; Soma "p1" com o parâmetro "p2"</p> <p>LDR B Soma ; Incrementa o ponteiro para retornar                      ADD B #1                      STR B Soma ; Ajusta o endereço de retorno</p> <p>JMP Soma,l</p>

Quanto ao acesso

Acesso Absoluto	Acesso Relativo
<ul style="list-style-type: none"> <li>• A referência (ponteiro) indica a posição de memória do parâmetro</li> <li>• O valor a ser processado é passado em endereço absoluto (um "parâmetro")</li> </ul>	<ul style="list-style-type: none"> <li>• A referência (ponteiro) indica a posição do parâmetros, a partir de uma posição de referência</li> <li>• Ver valor passado no registrador "X"</li> </ul>
<p>LDR A #vetor ; RA = vetor[0] + vetor[1]                      STR A param                      JSR SAbs</p> <p>LDR A #vetor+5 ; RA = vetor[5] + vetor[6]                      STR A param                      JSR SAbs</p> <p>SAbs: NOP                      LDR A param,l ; "param" é um ponteiro pro vetor</p> <p>LDR B param ; Incrementa o ponteiro                      ADD B #1                      STR B param</p> <p>ADD A param,l ; RA = vetor[i] + vetor[i+1]                      JMP SAbs,l</p>	<p>LDR X #0 ; RA = vetor[0] + vetor[1]                      JSR SRel</p> <p>LDR X #5 ; RA = vetor[5] + vetor[6]                      JSR SRel                      HLT</p> <p>SRel: NOP                      LDR A vetor,X ; Carrega em RA o vetor[x]                      ADD X #1 ; Incrementa o índice X do vetor                      ADD A vetor,X ; Soma o RA com o vetor[x]                      JMP SRel,l</p> <p>SRel2: NOP ; Alternativa sem incrementar o X                      LDR A vetor,X ; Carrega em RA o vetor[x]                      ADD A vetor+1,X ; Soma o RA com o vetor[x+1]                      JMP SRel2,l</p>

Tabela ASCII

		H0?	H1?	H2?	H3?	H4?	H5?	H6?	H7?
		x000	x001	x010	x011	x100	x101	x110	x111
H?0	0000	null	dle		0	@	P	`	p
H?1	0001	soh	dc1	!	1	A	Q	a	q
H?2	0010	stx	dc2	"	2	B	R	b	r
H?3	0011	etx	dc3	#	3	C	S	c	s
H?4	0100	eot	dc4	\$	4	D	T	d	t
H?5	0101	enq	nak	%	5	E	U	e	u
H?6	0110	ack	syn	&	6	F	V	f	v
H?7	0111	bell	etb	'	7	G	W	g	w
H?8	1000	bsp	can	(	8	H	X	h	x
H?9	1001	ht	em	)	9	I	Y	i	y
H?A	1010	lf	sub	*	:	J	Z	j	z
H?B	1011	vt	esc	+	;	K	[	k	{
H?C	1100	ff	fs	,	<	L	\	l	
H?D	1101	cr	gs	-	=	M	]	m	}
H?E	1110	so	rs	.	>	N	^	n	~
H?F	1111	si	us	/	?	O		o	del