



Arquitetura de Computadores

Conjunto de instruções do CESAR16 e CESAR16i

Instruções de uso geral			
H	Código	Instrução	Operação
0x	0000 xxxx	NOP	Nenhuma operação
Fx	1111 xxxx	HLT	Para a execução

Instruções sobre os códigos de condição (flags)			
H	Código	Instrução	Operação
1?	0001 nzvc	CCC NZVC	Clear Condition Code
2?	0010 nzvc	SCC NZVC	Set Condition Code

Instruções de controle de fluxo					
H	Código	Destino	Instrução	Operação	Condição
30	0011 0000	dd8	BR dd8	$R7 \leftarrow R7 + dd8$	sempre
31	0011 0001	dd8	BNE dd8	if (NotEqual) then $R7 \leftarrow R7 + dd8$	$z = 0$
32	0011 0010	dd8	BEQ dd8	if (Equal) then $R7 \leftarrow R7 + dd8$	$z = 1$
33	0011 0011	dd8	BPL dd8	if (Plus) then $R7 \leftarrow R7 + dd8$	$n = 0$
34	0011 0100	dd8	BMI dd8	if (Minus) then $R7 \leftarrow R7 + dd8$	$n = 1$
35	0011 0101	dd8	BVC dd8	if (Overflow Clear) then $R7 \leftarrow R7 + dd8$	$v = 0$
36	0011 0110	dd8	BVS dd8	if (Overflow Set) then $R7 \leftarrow R7 + dd8$	$v = 1$
37	0011 0111	dd8	BCC dd8	if (Carry Clear) then $R7 \leftarrow R7 + dd8$	$c = 0$
38	0011 1000	dd8	BCS dd8	if (Carry Set) then $R7 \leftarrow R7 + dd8$	$c = 1$
39	0011 1001	dd8	BGE dd8	(\geq) if (Greater or Equal) then $R7 \leftarrow R7 + dd8$	$n = v$
3A	0011 1010	dd8	BLT dd8	$(<)$ if (Less Than) then $R7 \leftarrow R7 + dd8$	$n <> v$
3B	0011 1011	dd8	BGT dd8	$(>)$ if (Greater Than) then $R7 \leftarrow R7 + dd8$	$n = v$ and $z = 0$
3C	0011 1100	dd8	BLE dd8	(\leq) if (Less or Equal) then $R7 \leftarrow R7 + dd8$	$n <> v$ or $z = 1$
3D	0011 1101	dd8	BHI dd8	$(x >)$ if (Higher) then $R7 \leftarrow R7 + dd8$	$c = 0$ and $z = 0$
3E	0011 1110	dd8	BLS dd8	$(x \leq)$ if (Lower or Same) then $R7 \leftarrow R7 + dd8$	$c = 1$ or $z = 1$
4x	0100 xxxx	xxmmmmrrr	JMP end	$R7 \leftarrow \text{end}$	sempre
5?	0101 xrrr	dd8	SOB reg,dd8	$\text{reg} \leftarrow \text{reg} - 1$; if $(\text{reg} <> 0)$ then $R7 \leftarrow R7 - dd8$	$R_i - 1 <> 0$
6?	0110 xrrr	xxmmmmrrr	JSR reg,end	$\text{Pilha} \leftarrow \text{reg}$; $\text{reg} \leftarrow R7$; $R7 \leftarrow \text{end}$	sempre
7?	0111 xrrr	-	RTS reg	$R7 \leftarrow \text{reg}$; $\text{reg} \leftarrow \text{Pilha}$	sempre

Instruções com 2 operandos					Flags			
H	Código	Operandos	Instrução	Operação	N	Z	V	C
9?	1001 mmmr	rrmmmmrrr	MOV src_end, dst_end	$\text{dst} \leftarrow \text{src}$	t	t	0	-
A?	1010 mmmr	rrmmmmrrr	ADD src_end, dst_end	$\text{dst} \leftarrow \text{dst} + \text{src}$	t	t	t	t
B?	1011 mmmr	rrmmmmrrr	SUB src_end, dst_end	$\text{dst} \leftarrow \text{dst} - \text{src}$	t	t	t	not(t)
C?	1100 mmmr	rrmmmmrrr	CMP src_end, dst_end	$\text{src} - \text{dst}$	t	t	t	not(t)
D?	1101 mmmr	rrmmmmrrr	AND src_end, dst_end	$\text{dst} \leftarrow \text{dst} \text{ AND } \text{src}$	t	t	0	-
E?	1110 mmmr	rrmmmmrrr	OR src_end, dst_end	$\text{dst} \leftarrow \text{dst} \text{ OR } \text{src}$	t	t	0	-

Instruções com 1 operando					Flags			
<i>H</i>	<i>Código</i>	<i>Operando</i>	<i>Instrução</i>	<i>Operação</i>	<i>N</i>	<i>Z</i>	<i>V</i>	<i>C</i>
80	1000 0000	xxmmmmrrr	CLR op	op ← 0	0	1	0	0
81	1000 0001	xxmmmmrrr	NOT op	op ← NOT op	t	t	0	1
82	1000 0010	xxmmmmrrr	INC op	op ← op + 1	t	t	t	t
83	1000 0011	xxmmmmrrr	DEC op	op ← op - 1	t	t	t	not(t)
84	1000 0100	xxmmmmrrr	NEG op	op ← - op	t	t	t	not(t)
85	1000 0101	xxmmmmrrr	TST op	op ← op	t	t	0	0
86	1000 0110	xxmmmmrrr	ROR op	Rotate “op” one bit right through carry	t	t	xor	lsb
87	1000 0111	xxmmmmrrr	ROL op	Rotate “op” one bit left through carry	t	t	xor	msb
88	1000 1000	xxmmmmrrr	ASR op	Shift “op” one bit right	t	t	xor	lsb
89	1000 1001	xxmmmmrrr	ASL op	Shift “op” one bit left	t	t	xor	msb
8A	1000 1010	xxmmmmrrr	ADC op	op ← op + c (carry)	t	t	t	t
8B	1000 1011	xxmmmmrrr	SBC op	op ← op - c (carry)	t	t	t	t

rrr ou reg - indica um dos 8 registradores (R0 a R7);

mmm - indica um dos 8 modos de endereçamento;

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

dd8 - representa um endereço de 8 bits;

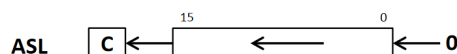
t - indica que o código de condição é testado pela unidade de controle e ajustado de acordo;

not(t) - para o carry indica que o valor carregado é o borrow-out;

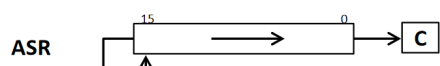
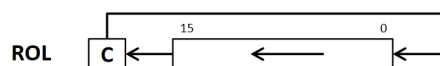
xor - para o overflow indica que este é carregado com o “ou exclusivo” entre os bits N e C após a execução da instrução;

Isb - é o bit menos significativo do operando (antes da execução);

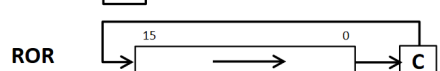
msb - é o bit mais significativo do operando (antes da execução);



Endereçamento do Visor



Endereçamento do Teclado



0FFDAH	Se o conteúdo deste endereço de memória for 080H, então existe tecla disponível
0FFDBH	Endereço a ser lido, caso exista uma tecla disponível

Modos de endereçamento (mmm)			
Código	Nome	Símbolo	Operação
000	Registrador	Ri	Ri
001	Reg. Pós-incrementado	(Ri)+	MEM(Ri); Ri + = 2
010	Reg. Pré-decrementado	-(Ri)	Ri - = 2; MEM(Ri)
011	Indexado	dd16(Ri)	MEM(dd16+Ri)
100	Reg. <i>Indireto</i>	(Ri)	MEM(Ri)
101	Pós-incrementado <i>Indireto</i>	((Ri)+)	MEM(MEM(Ri)); Ri + = 2
110	Pré-decrementado <i>Indireto</i>	(-(Ri))	Ri - = 2; MEM(MEM(Ri))
111	Indexado <i>Indireto</i>	(dd16(Ri))	MEM(MEM(dd16+Ri))

Registradores (rrr)	
Código	Reg.
000	R0
001	R1
010	R2
011	R3
100	R4
101	R5
110	R6 (SP)
111	R7 (PC)

Modos de endereçamento derivados utilizando o R7 (PC)				
Código	Nome	Instrução	Simbolo	Operação
xx001 111	Imediato	#dd16	(R7) + dd16	MEM(R7); R7 += 2
xx101 111	Absoluto	dd16	((R7)+)dd16	MEM(MEM(R7)); R7 += 2
xx011 111	Relativo	-	dd8(R7)	MEM(dd8 + R7)
xx111 111	Relativo Indireto	-	(dd8(R7))	MEM(MEM(dd8 + R7))

Dec.	Hex.	Endereços Especiais
65408	FF80	Endereço sugerido para o fundo da stack
65469	FFBD	31 palavras reservadas para uso futuro
65470	FFBE	IVET - Primeiro byte do “vetor de interrupção” - endereço da ISR (Interrupt Service Routine)
65471	FFBF	IVET - Segundo byte do “vetor de interrupção” - endereço da ISR
65472	FFC0	Início da área de acesso em 8 bits
65473	FFC1	23 bytes reservados para uso futuro
65495	FFD7	TIMDT - Timer base time
65496	FFD8	INTS - Interrupt Status - Monitoramento do estado das interrupções Bit 7: IP - Interrupt Pending - em geral só deve estar ligado quando executando a ISR Bit 1: IPStec - Interrupt Pending Source 1: Teclado Bit 0: IPStim - Interrupt Pending Source 0: Timer
65497	FFD9	INTE - Interrupt Enable - Controle de habilitação das interrupções Bit 7: IE - Interrupt Enable Bit 1: IESec - Interrupt Enable Source 1: Teclado Bit 0: IESim - Interrupt Enable Source 0: Timer
65498	FFDA	TECST - Se o conteúdo deste endereço de memória for H80 então existe tecla disponível
65499	FFDB	TECDT - Dado do Teclado - Endereço a ser lido caso exista uma tecla disponível
65500	FFDC	Início do endereçamento dos 36 bytes do visor
65535	FFFF	Fim do endereçamento do visor e fim da memória

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