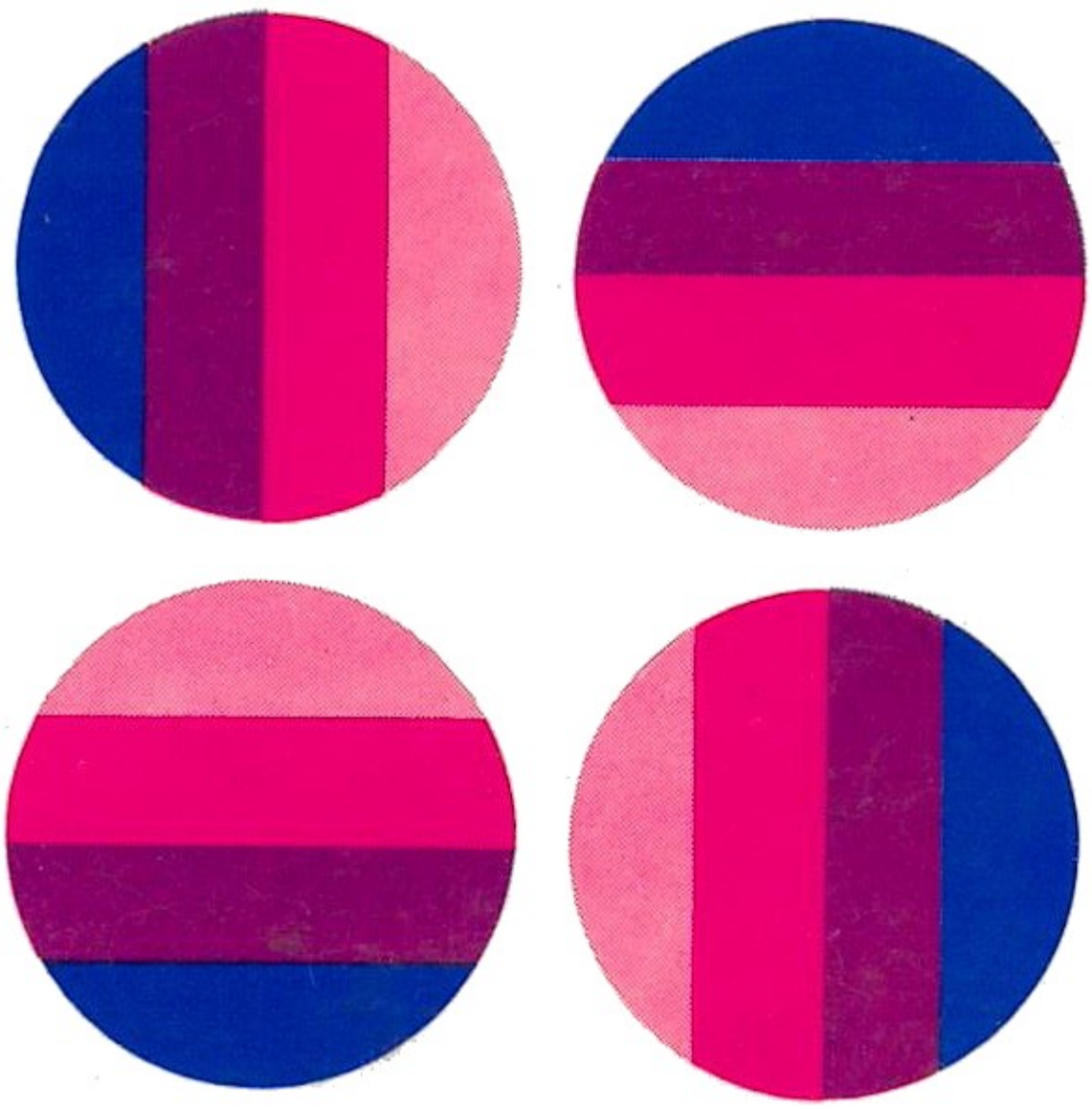
TRS COLOR

GUIA DE REFERÊNCIA

ROBERTO VALOIS



Editora Campus

TRS-80 COLOR

GUIA DE REFERÊNCIA

TRS COLOR

GUIA DE REFERÊNCIA

ROBERTO VALOIS

Editora Campus Ltda.

Rio de Janeiro

Capa
Otavio Studart

Projeto Gráfico, Composição e Revisão
Editora Campus Ltda.
Qualidade internacional a serviço do autor e do leitor nacional
Rua Barão de Itapagipe 55 Rio Comprido
Telefone: (021) 284-8443 Telex (00038) 021-32606
20261 Rio de Janeiro RJ Brasil
Endereço Telegráfico: CAMPUSRIO

ISBN 85-7001-366-3

Ficha Catalográfica
CIP-Brasil. Catalogação-na-fonte.
Sindicato Nacional dos Editores de Livros, RJ.

V283t

Valois, Roberto
TRS-Color; guia de referência / Roberto Valois. — Rio de Janeiro: Campus, 1986.

ISBN 85-7001-366-3

1. BASIC (Linguagem de programação para computadores) — Guias. 2. TRS-COLOR (Microcomputadores) — Guias. 3. ASSEMBLY (Linguagem de programação para computadores) — Guias. I. Título.

86-0717

CDD — 001.6424
CDU — 800.92BASIC

INTRODUÇÃO

Cada profissional precisa de um determinado tipo de ferramenta.

O programador não é uma exceção. Ele necessita de informações confiáveis sobre a máquina que utiliza. Estas devem estar dispostas de maneira lógica, para que a consulta seja fácil. Devem estar colocadas de maneira compacta, para que não seja necessário usar um volumoso manual. É necessário ainda que tenham sido organizadas por outro programador, que já tenha passado pelos mesmos problemas um dia.

Para atender a tudo isto, a Editora CAMPUS lança a Série GUIA DE REFERÊNCIA, escrita por brasileiros, para usuários de máquinas brasileiras.

Série GUIA DE REFERÊNCIA — a ferramenta do programador.

SUMÁRIO

1

Mapa de Memória

1. ÁREA DE COMUNICAÇÃO DO BASIC

0003 – números de subscritos no array
0006 – tipo de variável
= 0 – numérica
≠ 0 – string
0019 001A – início da área de programa BASIC
001B 001C – início da área de variáveis
001D 001E – início da área de array
001F 0020 – início da área de memória livre
0021 0022 – início da área de string
0023 0024 – próxima área livre para string
0027 0028 – área de memória reservada
0029 002A – BREAK, número da linha
002D 002E – BREAK, ESP
002F 0030 – último byte executado
0031 0032 – READ, número da linha
0033 0034 – READ, ponteiro
0037 0038 – nome da variável durante localização/criação
37 – 1.º caractere
38 – 2.º caractere – se string, bit 7 setado
004F 0054 – acumulador de ponto flutuante n.º 1
005C 0061 – acumulador de ponto flutuante n.º 2
0068 0069 – linha de BASIC em execução
006C – posição da linha em execução
006D – comprimento da linha
006F – número do periférico
0 = vídeo ou teclado –1 = cassete –2 = impressora
0070 – EOF
0071 – reset-flag
0072 0073 – reset-endereço
0074 0075 – final da RAM
0078 – OPEN/CLOSE
0 = cassete CLOSE 1 = cassete OPEN "I" 2 = cassete OPEN "0"
0079 – número de bytes no buffer do cassete
007A 007B – próxima locação livre no buffer do cassete
007C – tipo de bloco
0 = header 1 = dados FF = EOF

007D – cassete – comprimento do bloco
007E 007F – cassete buffer – endereço
0080 – cassete checksum
0081 – cassete código de erro
0082 – cassete – contador de bit
0083 – cassete – duração do bit
0087 – última tecla pressionada
0088 0089 – localização do cursor
008C – SOUND – tom
008D 008E – SOUND – duração
0094 – cor do cursor
0095 009C – variáveis da impressora
009D 009E – endereço do EXEC
009F 00AA – pega o próximo caractere (ESP)

```
INC < A7
BNE < A5
INC < A6
LDA ESP
JMP AA1A
```

00B5 – cor atual
00B6 – PMODE atual
00B7 00B8 – fim de área de gráficos
00B9 – número de bytes por linha
00BD – coordenada X1
00BF – coordenada Y1
00C3 – coordenada X2
00C5 – coordenada Y2
00EA 00EF – disk I/O
0100 0102 – SWI3
0103 0105 – SWI2
0106 0108 – SWI1
0109 010B – NMI
010C 010E – IRQ
010F 0111 – FIRQ
0112 0114 – USR
0116 0119 – RND – semente
011A – teclado flag SHIFT
011D 011F – desvio para EXP
0120 – número de comandos Color BASIC
0121 0122 – Color BASIC – comandos
0123 0124 – endereço da tabela de Jumps – comandos
0125 – número de funções Color BASIC
0126 0127 – Color BASIC – funções
0128 0129 – endereço da tabela de Jumps – funções
012A – número de comandos no Estendido Color BASIC

012B 012C – Estendido Color BASIC – comandos
012D 012E – endereço da tabela de Jumps no Estendido Color BASIC – comandos
012F – número de funções no Estendido Color BASIC
0130 0131 – Estendido Color BASIC – funções
0132 0133 – endereço da tabela de Jumps no Estendido Color BASIC – funções
0134 – número de comandos no Disk ECB
0135 0136 – Disk ECB – comandos
0137 0138 – endereço da tabela de Jumps no Disk ECB – comandos
0139 – número de funções no Disk ECB
013A 013B – Disk ECB – funções
013C 013D – endereço da tabela de Jumps no Disk ECB – funções
0152 0159 – teclado – área de trabalho

	FE	FD	FB	F7	EF	DF	BF
152	@	H	P	X	0	8	ENTER
153	A	I	Q	Y	1	9	CLEAR
154	B	J	R	Z	2	:	BREAK
155	C	K	S	↑	3	;	
156	D	L	T	↓	4	,	
157	E	M	U	←	5	–	
158	F	N	V	→	6	.	
159	G	O	W	ESPAÇO	7	/	

015A – joystick 0 posição X esquerda
015B – joystick 0 posição Y esquerda
015C – joystick 1 posição X direita
015D – joystick 1 posição Y direita
01D1 – comprimento do nome no buffer de nome do cassete
01D2 01D9 – buffer de nome de arquivo do cassete
01DA 02D9 – buffer – cassete
02DD 03DC – buffer – input
03DD 03FF – ASCII – buffer

2. COMANDOS DO ESTENDIDO COLOR BASIC

Endereço	Conteúdo	Comando
81F0	8970	DEL
81F2	8533	EDIT
81F4	86A7	TRON
81F6	86A8	TROFF
81F8	8871	DEF
81FA	AF89	LET
81FC	93BB	LINE
81FE	9532	PCLS
8200	9361	PSET
8202	9365	PRESET
8204	9670	SCREEN

8206	968B	PCLEAR
8208	9546	COLOR
820A	9E9D	CIRCLE
820C	98EC	PAINT
820E	9755	GET
8210	9758	PUT
8212	9CB6	DRAW
8214	9723	PCOPY
8216	9621	PMODE
8218	9A22	PLAY
821A	8C18	DLOAD
821C	8A09	RENUM

3. FUNÇÕES DO ESTENDIDO COLOR BASIC

Endereço	Conteúdo	Função
8257	83B0	ATN
8259	8378	COS
825B	8381	TAN
825D	84F2	EXP
825F	8524	FIX
8261	8446	LOG
8263	86AC	POS
8265	8480	SQR
8257	8BDD	HEX$
8269	86BE	VARPTR
826B	877E	INSTR
826D	8968	TIMER
826F	9339	PPOINT
8271	874E	STRING$

4. COMANDOS DO COLOR BASIC

Endereço	Conteúdo	Comando
AB67	AD47	FOR
AB69	AE86	GO
AB6B	AEE3	REM
AB6D	AEE3	'
AB6F	AEE3	ELSE
AB71	AF14	IF
AB73	AEE0	DATA
AB75	B8F7	PRINT
AB77	AF42	ON
AB79	AFF5	INPUT
AB7B	AE02	END
AB7D	B0F8	NEXT

AB7F	B34E	DIM
AB81	B046	READ
AB83	AE75	RUN
AB85	ADE4	RESTORE
AB87	AEC0	RETURN
AB89	AE09	STOP
AB8B	B757	POKE
AB8D	AE30	CONT
AB8F	B764	LIST
AB91	AE41	CLEAR
AB93	AD17	NEW
AB95	A498	CLOAD
AB97	A44C	CSAVE
AB99	A5F6	OPEN
AB9B	A416	CLOSE
AB9D	B75E	LLIST
AB9F	A880	SET
ABA1	A8B1	RESET
ABA3	A910	CLS
ABA5	A7BD	MOTOR
ABA7	A94B	SOUND
ABA9	A990	AUDIO
ABAB	A53E	EXEC
ABAD	A5EC	SKIPF

5. FUNÇÕES DO COLOR BASIC

Endereço	Conteúdo	Função
AA29	BC7A	SGN
AA2B	BCEE	INT
AA2D	BC93	ABS
AA2F	0112	USR
AA31	BF1F	RND
AA33	BF78	SIN
AA35	B750	PEEK
AA37	B681	LEN
AA39	B4FD	STR$
AA3B	B716	VAL
AA3D	B6A0	ASC
AA3F	B68C	CHR$
AA41	A5CE	EOF
AA43	A9C6	JOYSTK
AA45	B6AB	LEFT$
AA47	B6C8	RIGHT$
AA49	B6CF	MID$

AA4B	A8F5	POINT
AA4D	A564	INKEY$
AA4F	B4EE	MEM

6. INPUT — OUTPUT

a) PIA 1(A) FF00 (Registrador de dados)

Bit	Função
0	teclado — linha 1 e botão do joystick direito
1	teclado — linha 2 e botão do joystick esquerdo
2	teclado — linha 3
3	teclado — linha 4
4	teclado — linha 5
5	teclado — linha 6
6	teclado — linha 7
7	joystick — entrada para o comparador

b) PIA 1(A) FF01 (Registrador de controle)

Bit	Função
0	63.5 μs IRQ (1 = habilitado)
1	63.5 μs IRQ polaridade
2	FF00 — dados / direção (1 = dados)
3	bit menos significativo do multiplexador analógico
4	sempre 1
5	sempre 1
6	não usado
7	63.5 μs IRQ (1 = transição no sinal)

c) PIA 1(B) FF02 (Registrador de dados)

Bit	Função
0	teclado — coluna 1
1	teclado — coluna 2
2	teclado — coluna 3
3	teclado — coluna 4
4	teclado — coluna 5
5	teclado — coluna 6
6	teclado — coluna 7
7	teclado — coluna 8

d) PIA 1(B) FF03 (Registrador de controle)

Bit	Função
0	16.7 ms IRQ (1 = habilitado)
1	16.7 ms IRQ polaridade

2	FF02 – dados / direção (1 = dados)
3	bit menos significativo do multiplexador analógico
4	sempre 1
5	sempre 1
6	não usado
7	16.7 ms IRQ (1 = transição no sinal)

e) PIA 2(A) FF20 (Registrador de dados)

Bit	Função
0	Cassete – entrada de dados (1 = entrada < –1.5V)
1	RS-232 – saída de dados (1 = saída em –10V)
2 a 7	Digital para analógico

f) PIA 2(A) FF21 (Registrador de controle)

Bit	Função
0	Detecção da portadora da RS-232 via FIRQ (1 = habilitado)
1	Polaridade da detecção da portadora da RS-232
2	FF20 – dados / direção (1 = dados)
3	Cassete – controle do motor (1 = ON)
4	sempre 1
5	sempre 1
6	não usado
7	Detecção da portadora (1 = transição de sinal)

g) PIA 2(B) FF22 (Registrador de dados)

Bit	Função
0	entrada de dados da RS-232 (1 = entrada < +1V)
1	saída de 1 bit para gerador de som (normalmente zero)
2	entrada do tamanho da RAM (0 = 4K 1 = 16K alta impedância = 64K)
3	seleção de cores do VDG
4	VDG GM0 e INT/EXT
5	VDG GM1
6	VDG GM2
7	VDG gráfico/alfa (1 = gráfico)

h) PIA 2(B) FF23 (Registrador de controle)

Bit	Função
0	detecção de interrupção do cartucho via FIRQ (1 = habilitado)
1	polaridade da detecção de interrupção do cartucho (normalmente 0)
2	FF20 dados/direção (1 = dados)
3	geração de som por 6 bits (1 = habilitado)
4	sempre 1
5	sempre 1

6	não usado
7	detecção da interrupção do cartucho (1 = transição de sinal)

7. REGISTRADORES GERAIS

FF48	registro status/comando
FF49	registrador de trilha
FF4A	registrador de setor
FF4B	registrador de dados

8. REGISTRADOR DE CONTROLE-SAM

Desativado (0)	Ativado (1)		
FFDE	FFDF	(TY)	tipo de mapa (0 = 32K RAM, 1 = 64K RAM)
FFDC	FFDD	(M0)	tipo de memória
FFDA	FFDB	(M1)	00 = 4K, 01 = 16K, 10 = 64K
FFD8	FFD9	(R1)	taxa do clock
FFD6	FFD7	(R0)	00 = 0.9MHz, 01 = dependente de endereço
FFD4	FFD5	(P1)	página de memória (0 = paginado 1 = página 1)
FFD2	FFD3	(F6)	endereço inicial da tela
FFD0	FFD1	(F5)	
FFCE	FFCF	(F4)	
FFCC	FFCD	(F3)	
FFCA	FFCB	(F2)	
FFC8	FFC9	(F1)	
FFC6	FFC7	(F0)	
FFC4	FFC5	(V2)	modo de vídeo
FFC2	FFC3	(V1)	
FFC0	FFC1	(V0)	

9. MODO DE CONTROLE DO VÍDEO

	FF22					FFC5	FFC3	FFC1
X Y	COR	7A/G	6GM2	5GM1	4GM0	V2	V1	V0
256x192	2	1	1	1	1	1	1	0
128x192	4	1	1	1	0	1	1	0
128x192	2	1	1	0	1	1	0	1
128x96	4	1	1	0	0	1	0	0
128x96	2	1	0	1	1	0	1	1
128x64	4	1	0	1	0	0	1	0
128x64	2	1	0	0	1	0	0	1
64x64	4	1	0	0	0	0	0	1

ALPHA								
64x32	8	0	0	0	0	0	0	0
64x48	4	0	0	0	1	0	0	0
64x64	8	0	0	0	0	0	1	0
64x96	8	0	0	0	0	1	0	0
64x192	8	0	0	0	0	1	1	0

10. DISPOSIÇÃO DAS VARIÁVEIS NA MEMÓRIA

a/ Variável Numérica

00	primeiro caractere do nome da variável
+01	segundo caractere do nome da variável
+02	expoente
+03	byte mais significativo
+04	segundo byte mais significativo
+05	terceiro byte mais significativo
+06	byte menos significativo

b/ Variável String

00	primeiro caractere do nome da variável
+01	segundo caractere do nome da variável com bit 7 setado
+02	comprimento da string
+03	reservado
+04	byte mais significativo do endereço da string
+05	byte menos significativo do endereço da string
+06	reservado

c/ Variável Array

00	primeiro caractere do nome da variável
+01	segundo caractere do nome da variável
+02	byte mais significativo do endereço do próximo array
+03	byte menos significativo do endereço do próximo array
+04	número de subscritos no array
+05	byte mais significativo do número de elementos +1 no 2.º subscrito
+06	byte menos significativo do n.º de elementos +1 no 2.º subscrito
+07	byte mais significativo do n.º de elementos +1 no 1.º subscrito
+08	byte menos significativo do n.º de elementos +1 no 1.º subscrito
+09	5 bytes para cada elemento preenchendo coluna por coluna
.	
.	
.	

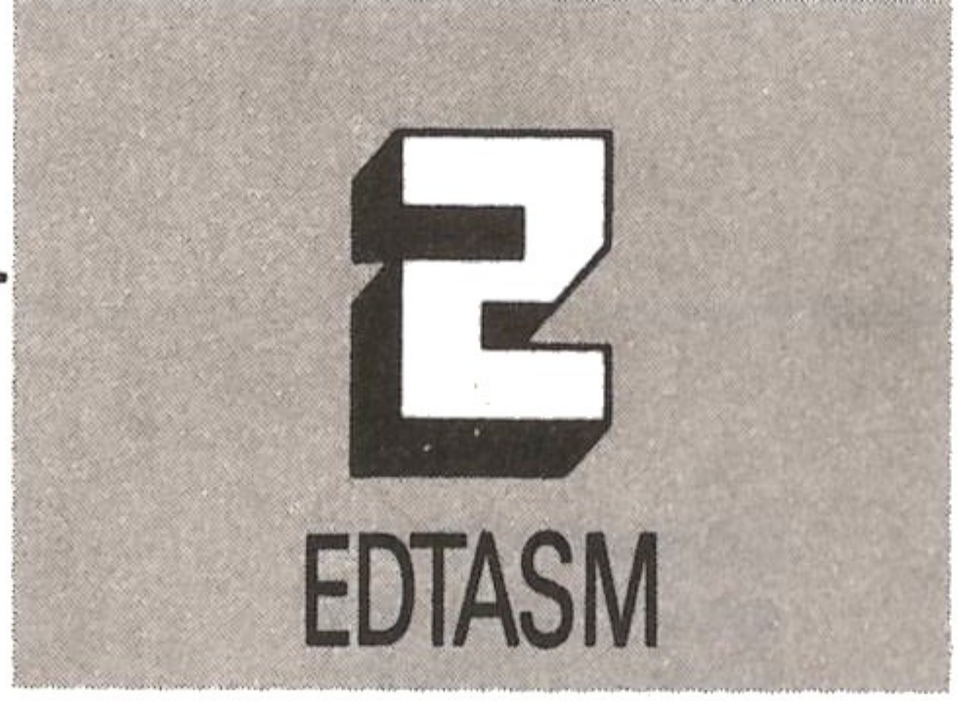
11. DISPOSIÇÃO DOS COMANDOS FOR E GOSUB

a/ FOR

00	byte mais significativo do ESP do FOR
–01	byte menos significativo do ESP do FOR
–02	byte mais significativo do n.º da linha BASIC do FOR
–03	byte menos significativo do n.º da linha BASIC do FOR
–04	expoente do valor do TO
–05	byte mais significativo do valor do TO
–06	segundo byte mais significativo do valor do TO
–07	terceiro byte mais significativo do valor do TO
–08	byte menos significativo do valor do TO
–09	sinal do flag do valor do STEP
–0A	expoente do valor do STEP
–0B	byte mais significativo do valor do STEP
–0C	segundo byte mais significativo do valor do STEP
–0D	terceiro byte mais significativo do valor do STEP
–0E	byte menos significativo do valor do STEP
–0F	byte mais significativo do VARPTR do FOR
–10	byte menos significativo do VARPTR do FOR
–11	token do FOR

b/ GOSUB

00	byte mais significativo do endereço de retorno
–01	byte menos significativo do endereço de retorno
–02	byte mais significativo do ESP do GOSUB
–03	byte menos significativo do ESP do GOSUB
–04	byte mais significativo do n.º da linha BASIC do GOSUB
–05	byte menos significativo do n.º da linha BASIC do GOSUB
–06	token do GOSUB



1. EDITOR

A Assemble	"assembla texto"
C Copy	copia um bloco de linhas para nova posição. Ex.: C500,100:150,5 copia de 100 a 150 para 500 com incremento de 5
D Delete	deleta uma ou mais linhas. Ex.: D300:500 delete de 300 a 500
E Edit	entra no subcomando Edit. Ex.: E100 edita linha 100
F Find	procura string. Ex.: FABC procura a partir da linha atual a primeira ocorrência da string ABC
H Hardcopy	lista na impressora. Ex.: H300:500 lista as linhas de 300 a 500
I Insert	Insere linhas. Ex.: I300 insere a partir de 300 com incremento de 3
L Load	lê do cassete o programa fonte
M Move	move um bloco de linhas. Ex.: M1300,100:500,2 move de 100 a 500 para a posição iniciando em 1300 com incremento de 2
N Renumber	renumera linhas. Ex.: N100,10 renumera as linhas iniciando na 100 com incremento de 10
P Display	coloca na tela as linhas de programa. Ex.: P100:500 apresenta na tela as linhas de 100 a 500
Q Quit	volta ao BASIC
R Replace	troca a linha por uma nova e inicia inserção. Ex.: R100,3 troca a linha 100 por uma nova e inicia inserção com incremento de 3
T Print	o mesmo que H exceto que os nºs das linhas não são printados
V Verify	compara o conteúdo do programa de memória com o que foi previamente gravado
W Write	grava o buffer de texto no cassete. Ex.: W NOME
Z ZBUG	transfere o controle para ZBUG

2. SWITCH DO ASSEMBLY

/AO	ORG absoluto
/IM	Assemble na memória
/LP	Listando na impressora
/MO	ORG manual
/NL	Sem listagem
/NO	Sem código objeto
/NS	Sem tabela de símbolos
/WE	Pausa se existirem erros

3. ZBUG

A ASCII	apresenta em ASCII
B Bytemode	apresenta valor byte a byte
C Continue	continua após breakpoint
D Display	apresenta os breakpoints setados
E Editor	retorna ao editor
G Go	executa. Ex.: GA00
H Half symbolic	apresenta operandos em números
I Input base	seta base numérica de entrada. Ex.: I10 — seta decimal
L Load	lê programa em LM
M Mnemonic mode	as instruções são apresentadas em mnemônicos
N Numeric mode	apresenta sem símbolos
O Output base	seta base numérica de apresentação
P Save tape	salva memória no cassete: Ex.: PNOME 3000 (inicial) 4000 (final) 3000 (execução)
R Display register	mostra o conteúdo dos registradores
S Symbolic form	apresenta os endereços na forma simbólica
T Display block	apresenta a localização da memória correspondente às linhas. Ex.: T 1000 2000
TH Hardcopy block	idem na impressora
U Move block	transfere bloco. Ex.: M 3000 4000 10 move 16 bytes de 3000 a 300F para 4000 a 400F
V Verify	verifica se o programa em linguagem de máquina do cassete é igual ao da memória
W Word mode	apresenta memória no modo 16 bits (palavra)
X Breakpoint	seta breakpoint. Ex.: X400
Y Yak breakpoint	elimina breakpoint. Ex.: Y3 elimina breakpoint nº3

a — apresenta anterior
— apresenta próximo
; — força modo numérico
— força modo numérico e modo de byte
: — converte byte corrente para flag
/ — examina registros ou localização de memória
, — executa uma instrução (single step)

3

BASIC

1. INSTRUÇÕES DO BASIC

ABS(X)
retorna o valor absoluto de X

ASC(X$)
retorna o código ASCII do 1º caractere de X$

ATN(X)
retorna o arco tangente de X em radianos

AUDIO ON/OFF
conecta/desconecta saída do gravador com alto-falante da TV

CHR$(X)
retorna caractere cujo código ASCII é X

CIRCLE(X,Y),r,c,l,i,f
desenha elipse com centro em (X,Y), raio r, cor c, razão altura/largura l (0 < 1 < 255), ponto inicial i (0 < i < 1), ponto final f (0 < f < 1)

CLEAR X,Y
reserva X bytes a partir do endereço Y

CLOAD"NOME"
lê programa do cassete

CLOAD"NOME",num
carrega programa do cassete em linguagem de máquina no endereço em que foi salvo mais um

CLOSE disp
fecha arquivo ou dispositivo

CLS c
limpa a tela com cor c

COLOR c1,c2
define cor de 1º plano c1 e cor de fundo c2

CONT
continua execução de programa após BREAK

COS(X)
retorna o cosseno de X em radianos

CSAVE"nome" ,A
salva programa no gravador. Se opção A é utilizada o programa é salvo em ASCII

CSAVE"nome",i,f,t
salva arquivo em linguagem de máquina endereço inicial i, final f

DATA item {,item,item...}
armazena dados no programa

DEF FN nome(var) = fórmula
define função nome = fórmula com parâmetro var

DEFUSR n end
define endereço de entrada end para a função USR de número n (0 < n < 9)

DEL n1-n2
anula as linhas entre n1 e n2

DIM var(num {,num,num...})
define matrizes e dimensões

DLOAD "nome",n
carrega um programa na velocidade especificada (n = 0→300 baud, n = 1→1200 baud)

DRAW linha
desenha uma linha cujo início, comprimento e cor são especificados

EDIT n
permite edição da linha de nº n

END
termina o programa

EOF
retorna 0 se houver mais dados, –1 se fim de arquivo

EXEC end
transfere o controle para programa em LM a partir do endereço end

EXP(X)
retorna a exponencial de X

FIX(X)
retorna a parte inteira de X

FOR var n1 TO n2 STEP n3
cria loop com var variando de n1 a n2 com passo n3

GET (x1,y1)-(x2,y2),d,G
copia para a matriz d o retângulo cuja diagonal é definida por (x1,y1), (x2,y2)

GOSUB n
desvia o fluxo do programa para a sub-rotina com início na linha n

GOTO n
desvia o fluxo do programa para a linha n

HEX$(X)
retorna string contendo o valor hexadecimal de X

IF cond THEN a1 ELSE
a2 executa a ação a1 ou a2 dependendo da condição cond

INKEY$
retorna a última tecla pressionada

INPUT"string",v1,v2...
apresenta mensagem e recebe dados no teclado

INPUT # –1, ,a
recebe dados do gravador

INSTR(n1,s1,s2)
retorna a posição da ocorrência da string s2 em s1

INT(X)
retorna valor inteiro de X

JOYSTK(X)
retorna coordenada vertical ou horizontal do joystick direito ou esquerdo. Se
X 0 horizontal joystick esquerdo
X 1 vertical joystick esquerdo
X 2 horizontal joystick direito
X 3 vertical joystick direito

LEFT$(S,N)
retorna substring de comprimento N da parte esquerda da string S

LEN(X$)
retorna o comprimento da string X$

LET var valor
atribui valor à variável var

LIST n1-n2
lista linhas compreendidas entre n1 e n2 na tela

LLIST n1-n2
lista linhas compreendidas entre n1 e n2 na impressora

LINE(x1,y1)-(x2,y2),PSET,BF
desenha linha com ponto inicial (x1,y1) ponto final (x2,y2). PSET seleciona cor de 1.º plano, PRESET seleciona cor de fundo, a opção B desenha um retângulo cuja diagonal é definida por (x1,y1) e (x2,y2) e a opção BF pinta o retângulo com a cor de 1.º plano

LINE INPUT"string";var
apresenta mensagem e recebe linha de texto em var pelo teclado

LOG(X)
retorna o logaritmo natural de X

MEM
fornece quantidade de memória livre

MID$(S,N1,N2)
retorna substring com N2 caracteres a partir da posição N1

MOTOR ON/OFF
liga/desliga motor do gravador

NEW
limpa memória

ON i GOSUB n1,n2...
desvia para a iésima sub-rotina dependendo do valor de i (se i = 1 vai para n1, se i = 2 vai para n2...)

ON i GOTO n1,n2...
desvia para a iésima linha dependendo do valor de i

OPEN"I/O",disp,f
abre arquivo f no dispositivo disp, de entrada (I) ou saída (O)

PAINT (x,y),c1,c2
pinta gráfico na tela a partir do ponto x,y com cor c1 parando quando encontrar c2

PCLEAR N
RESERVA N PÁGINAS DE MEMÓRIA para tela gráfica

PCLS n
limpa a tela com cor n

PCOPY n1 to n2
copia gráfico da página gráfica n1 para n2

PEEK(end)
retorna o conteúdo do endereço de memória
PLAY s1
toca música especificada pela string s1
PMODE n1,n2
seleciona resolução n1 e página de memória inicial n2
POINT (x,y)
testa se o ponto x,y está aceso ou apagado
POKE n1,n2
armazena o valor n2 no endereço de memória n1
POS(n)
fornece a posição do cursor no dispositivo n
PPOINT(x,y)
fornece o código de cor do ponto especificado
PRESET(x,y)
coloca cor de fundo no ponto especificado
PRINT var
imprime var na tela da tv
PRINT disp,var
coloca var no dispositivo especificado
PRINT TAB(n)"string"
move o cursor para a coluna n e imprime string
PRINT USING "form";var
imprime var no formato especificado
PRINT § n,var
imprime var na posição da n-tela
PSET (x,y,c)
coloca um ponto de cor na posição x,y
PUT (x1,y1)-(x2,y2),m,a
coloca gráfico armazenado em uma matriz m na tela na posição definida pelo retângulo cuja diagonal é definida pelos pontos (x1,y1), (x2,y2). A ação a pode ser: PSET;PRESET;AND;OR;NOT
READ VAR
lê o item da linha DATA
REM ou '
permite a inserção de comentários na linha de programa
RENUM n1,n2,n3
renumera linhas de programa a partir da n2 com incremento n3 com as novas linhas iniciando em n1
RESET (x,y)
apaga o ponto x,y
RESTORE
coloca o indicador para o primeiro item na linha de DATA
RETURN
retorna da sub-rotina
RIGHT$(X$,n)
retorna substring da string X$ com n caracteres a partir da direita

RND(n)
retorna um número aleatório entre 1 e n

RUN n
inicia a execução do programa a partir da linha especificada

SCREEN n1,n2
seleciona tela gráfica ou de texto e conjunto de cores

SET(x,y,c)
coloca ponto x,y com cor c

SGN(n)
retorna o sinal da expressão numérica n –1 negativo, 0 se zero, 1 positivo

SKIPF "nome"
inicia leitura do gravador após o programa especificado

SIN(x)
retorna o seno de x em radianos

SOUND n1,n2
emite som com tom n1 e duração n2

STOP
interrompe execução do programa

STRING$(n1,n2)
retorna string de caracteres cujo código ASCII é igual a n2 de comprimento n1

STR$(x)
retorna a conversão de x em string

SQR(x)
retorna a raiz quadrada de x

TAN(x)
retorna a tangente de x

TIMER
fornece o conteúdo do temporizador e permite sua mudança

TROFF
desativa TRACE

TRON
ativa trace

USRn
chama a sub-rotina em LM n

VAL (X$)
retorna a conversão da string X$ em número

VARPTR(X)
retorna o endereço de memória da variável x

2. MENSAGENS DE ERRO

Abreviação	Explicação
/0	Divisão por zero
A0	Tentativa de abrir um arquivo que já foi aberto
BS	Índice inválido

CN	Não pode continuar
DD	Tentativa de redimensionar uma matriz
DN	Erro de número de dispositivo
DS	Instrução direta no arquivo de dados
FC	Chamada de função ilegal
FD	Dados de arquivo inválido
FM	Modo de arquivo inválido
ID	Instrução direta ilegal
IE	Entrada depois do final do arquivo
IO	Erro de entrada/saída
LS	String longa
NF	NEXT sem FOR
NO	Arquivo não aberto
OD	Faltam dados para READ
OM	Falta memória
OS	Falta espaço para strings
OV	Overflow
RG	RETURN sem GOSUB
SN	Erro de sintaxe
ST	Fórmula de string muito complexa
TM	Tipo errado
UL	Linha indefinida

3. EDIÇÃO DE PROGRAMA

EDIT	permite a edição de uma linha de programa
nC	altera n caracteres
nD	deleta n caracteres
I	entra em modo de inserção
H	anula o resto da linha e entra no modo de inserção
nKc	apaga todo o texto até a enésima ocorrência do caractere c
L	lista a linha atual e continua edição
nSc	procura a enésima ocorrência do caractere c
X	move o cursor para o final da linha e entra no modo de inserção
SHIFT	sai do modo de inserção
n	move o cursor n espaços para a esquerda
n ESPAÇO	move o cursor n espaços para a direita

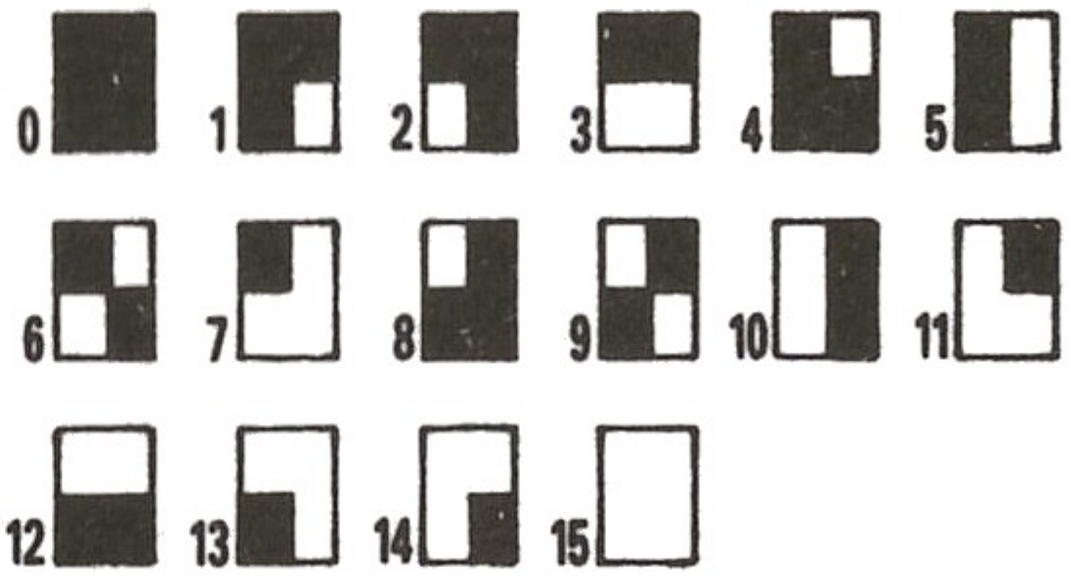
4

Vídeo

1. TABELA DOS CONJUNTOS DE CORES

PMODE n	SCREEN 1,c	DUAS CORES	QUATRO CORES
0	0	preto/verde	
	1	preto/cinza	
1	0		verde/amarelo/azul/vermelho
	1		cinza/ciano/roxo/laranja
2	0	preto/verde	
	1	preto/cinza	
3	0		verde/amarelo/azul/vermelho
	1		cinza/ciano/roxo/laranja
4	0	preto/verde	
	1	preto/cinza	

CÓDIGO	COR
0	Preto
1	Verde
2	Amarelo
3	Azul
4	Vermelho
5	Cinza
6	Ciano
7	Roxo
8	Laranja



***Código** = 112 + 16 * cor + caractere*

2. CÓDIGO DOS CARACTERES NO VÍDEO

Carac	ASCII dec	ASCII hex	Vídeo dec	Vídeo hex
space	32	20	96	60
!	33	21	97	61
"	34	22	98	62
#	35	23	99	63
$	36	24	100	64
%	37	25	101	65
&	38	26	102	66
'	39	27	103	67
(	40	28	104	68
)	41	29	105	69
*	42	2A	106	6A
+	43	2B	107	6B
,	44	2C	108	6C
-	45	2D	109	6D
.	46	2E	110	6E
/	47	2F	111	6F
0	48	30	112	70
1	49	31	113	71
2	50	32	114	72
3	51	33	115	73
4	52	34	116	74
5	53	35	117	75
6	54	36	118	76
7	55	37	119	77
8	56	38	120	78
9	57	39	121	79
:	58	3A	122	7A
;	59	3B	123	7B
<	60	3C	124	7C
=	61	3D	125	7D
>	62	3E	126	7E
?	63	3F	127	7F
@	64	40	64	40
A	65	41	65	41
B	66	42	66	42
C	67	43	67	43
D	68	44	68	44
E	69	45	69	45
F	70	46	70	46
G	71	47	71	47
H	72	48	72	48
I	73	49	73	49
J	74	4A	74	4A

K	75	4B	75	4B
L	76	4C	76	4C
M	77	4D	77	4D
N	78	4E	78	4E
O	79	4F	79	4F
P	80	50	80	50
Q	81	51	81	51
R	82	52	82	52
S	83	53	83	53
T	84	54	84	54
U	85	55	85	55
V	86	56	86	56
W	87	57	87	57
X	88	58	88	58
Y	89	59	89	59
Z	90	5A	90	5A
[	91	5B	91	5B
\	92	5C	92	5C
]	93	5D	93	5D
↑	94	5E	94	5E
←	95	5F	95	5F
@ invertido	96	60	0	00
a	97	61	1	01
b	98	62	2	02
c	99	63	3	03
d	100	64	4	04
e	101	65	5	05
f	102	66	6	06
g	103	67	7	07
h	104	68	8	08
i	105	69	9	09
j	106	6A	10	0A
k	107	6B	11	0B
l	108	6C	12	0C
m	109	6D	13	0D
n	110	6E	14	0E
o	111	6F	15	0F
p	112	70	16	10
q	113	71	17	11
r	114	72	18	12
s	115	73	19	13
t	116	74	20	14
u	117	75	21	15
v	118	76	22	16
w	119	77	23	17
x	120	78	24	18

y	121	79	25	19
z	122	7A	26	1A
[invertido	123	7B	27	1B
/ invertido	124	7C	28	1C
] invertido	125	7D	29	1D
↑ invertido	126	7E	30	1E
← invertido	127	7F	31	1F

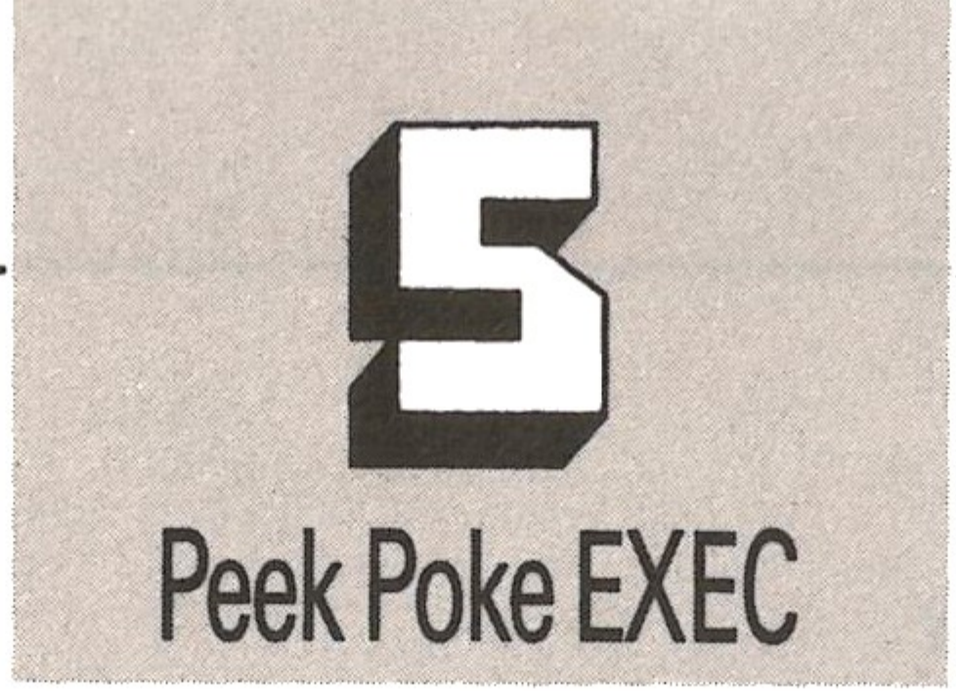
3. MODOS GRÁFICOS

Modo Gráfico	Resolução	Tamanho do Elemento	Código	Seleção de Cor 0	1	Memória Usada
6R	256x192		0	preto	preto	0400-1BFF
			1	verde	cinza	
6C	128x192		00	verde	cinza	0400-1BFF
			01	amarelo	ciano	
			10	azul	roxo	
			11	vermelho	laranja	
3R	128x192		0	preto	preto	0400-0FFF
			1	verde	cinza	
3C	128x96		00	verde	cinza	0400-0FFF
			01	amarelo	ciano	
			10	azul	roxo	
			11	vermelho	laranja	
2R	128x96		0	preto	preto	0400-09FF
			1	verde	cinza	
2C	128x64		00	verde	cinza	0400-0BFF
			01	amarelo	ciano	
			10	azul	roxo	
			11	vermelho	laranja	
1R	128x64		0	preto	preto	0400-07BF
			1	verde	cinza	
1C	64x64		00	verde	cinza	0400-07FF
			01	amarelo	ciano	
			10	azul	roxo	
			11	vermelho	laranja	

4. SAM E VDG

Modo Gráfico	Seleção de Cores	Instruções	
		Ativa	Desativa
6R	0	LDA #$F0 STA $FF22 STA $FFC3 STA $FFC5	LDA #0 STA $FF22 STA $FFC2 STA $FFC4
	1	LDA #$F8 STA $FF22 STA $FFC3 STA $FFC5	LDA #0 STA $FF22 STA $FFC2 STA $FFC4
6C	0	LDA #$E0 STA $FF22 STA $FFC3 STA $FFC5	LDA #0 STA $FF22 STA $FFC2 STA $FFC4
	1	LDA #$E8 STA $FF22 STA $FFC3 STA $FFC5	LDA #0 STA $FF22 STA $FFC2 STA SFFC4
3R	0	LDA #$D0 STA $FF22 STA $FFC1 STA $FFC5	LDA #0 STA $FF22 STA $FFC0 STA $FFC4
	1	LDA #$D8 STA $FF22 STA $FFC1 STA $FFC5	LDA #0 STA $FF22 STA $FFC0 STA $FFC4
3C	0	LDA #$C0 STA $FF22 STA $FFC5	LDA #0 STA $FF22 STA $FFC4
	1	LDA #$C8 STA $FF22 STA $FFC5	LDA #0 STA $FF22 STA $FFC4

Modo Gráfico	Seleção de Cores	Instruções	
		Ativa	Desativa
2R	0	LDA #$B0 STA $FF22 STA $FFC1 STA $FFC3	LDA #0 STA $FF22 STA $FFC0 STA $FFC2
	1	LDA #$B8 STA $FF22 STA $FFC1 STA $FFC3	LDA #0 STA $FF22 STA $FFC0 STA $FFC2
2C	0	LDA #SA0 STA $FF22 STA $FFC3	LDA #0 STA $FF22 STA $FFC2
	1	LDA #SA8 STA $FF22 STA $FFC3	LDA #0 STA $FF22 STA $FFC2
1R	0	LDA #$90 STA $FF22 STA $FFC1	LDA #0 STA $FF22 STA $FFC0
	1	LDA #$98 STA $FF22 STA $FFC1	LDA #0 STA $FF22 STA $FFC0
1C	0	LDA #$80 STA $FF22 STA $FFC1	LDA #0 STA $FF22 STA $FFC0
	1	LDA #$88 STA $FF22 STA $FFC1	LDA #0 STA $FF22 STA $FFC0



Peek Poke EXEC

1. GERAIS

PEEK(25)*256+PEEK(26)
endereço inicial do programa BASIC

PEEK(27)*256+PEEK(28)
endereço final do programa BASIC

POKE 113,0
desliga o botão de Reset (se reset pressionado — Coldstart)

POKE 113,85
liga o botão de Reset

PEEK(116)*256+PEEK(117)
máximo de memória do sistema

PEEK(485)*256+PEEK(486)
endereço de execução (EXEC) do programa em LM lido do cassete

PEEK(48)*256+PEEK(488)
endereço inicial do programa em LM lido do cassete

EXEC 42753
lê um bloco do cassete

EXEC 42954
o mesmo que MOTOR ON

EXEC 44326
apaga todas as variáveis e strings

POKE 65495,0
dobra a velocidade do clock

POKE 65494,0
volta a velocidade normal depois de POKE 65495,0

POKE 65497,0
triplica a velocidade do clock (perde imagem)

POKE 65496,0
volta a velocidade normal depois do POKE 65497,0

POKE 282,0
texto reverso

POKE 359,60
scroll lento

POKE 359,126
scroll normal

POKE 383,158
desabilita LIST

POKE 383,0
LIST normal
POKE 25,6:NEW
PCLEARO
POKE 25,14:3584,0:NEW
PCLEARO para disco
POKE 41382,128
cursor ciano
POKE 41382,192
cursor azul
POKE 44014,X:POKE 44015,X
troca do OK
POKE 178,X
novas cores
POKE 65314,0
SCREEN 0,0
POKE 65314,8
SCREEN 0,1
POKE 65314,192
cores 0
POKE 65314,200
cores 1
POKE 65314,248
preto, vermelho, branco, azul
POKE 65314,240
4 verdes
POKE 298,0:POKE 303,0
desabilita o BASIC ESTENDIDO
POKE 298,25:POKE 303,14
habilita o BASIC ESTENDIDO
POKE 65313,4
MOTOR ON
POKE 65313,52
MOTOR OFF
POKE 111,54:DIR
cópia do diretório
EXEC 41175
revisão da ROM
EXEC 44539
pausa até tecla ser pressionada

2. RS-232

Baud	POKE 149	POKE 150
50	4	88
75	2	227
110	1	246
134,5	1	153
150	1	110
300		180
600		87
1200		40
1800		25
2000		23
2400		18
3600		10
4800		7
7200		3
9600		1

6 Assembly

1. REGISTRADORES

8 bits A e B – Registradores de uso geral
16 bits D – Combinação de A e B
8 bits DP – (Direct Page) – guarda o byte mais significativo (MSB) do endereço de 16 bits
16 bits X e Y – Utilizado normalmente como Index
16 bits PC – Contador de programa
16 bits U – Ponteiro do stack do usuário
16 bits S – Ponteiro do stack do sistema
8 bits CC – Flags

- bit 0 – Carry
- 1 – Overflow
- 2 – Zero
- 3 – Negative
- 4 – Interrupt Request Mask
- 5 – Half Carry
- 6 – Fast Interrupt Request Mask
- 7 – Entire Flag

2. MODOS DE ENDEREÇAMENTO

1 – Inerente – sem operando. Ex. CLRA

2 – Imediato – o operando é um dado, o caractere # identifica o modo. Ex. CMPX #$1234

3 – Estendido – o operando é um endereço. Ex. LDD $1234. Estendido indireto – o operando é um endereço de um endereço. Ex. LDD [$1234]

4 – Indexado – o operando é um índice que aponta para um endereço. O caractere "," indica modo indexado. Ex. STA ,X. Indexado indireto – o operando é um índice que aponta para um endereço de um endereço. Ex. STA [,X]

5 – Relativo – o operando é um endereço relativo. O processador utiliza automaticamente este modo nas instruções de BRANCH. Ex. BRA SO585

6 – Direto – o operando é meio endereço. A outra metade é o conteúdo do registrador DP (MSB).

3. QUADRO DE INSTRUÇÕES

Modo de Endereçamento →	Inerente			Imediato			Direto			Extend			Indexado/ Indireto			Relativo			
Forma do Operando →				dado 8 ou 16 bits			end8			end16			ver Tab			Label ou desl.			
Mnemônico ↓	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Obs.
ABX	3A	3	1																
ADCA				89	2	2	99	4	2	B9	5	3	A9	4+	2+				
ADCB				C9	2	2	D9	4	2	F9	5	3	E9	4+	2+				
ADDA				8B	2	2	9B	4	2	BB	5	3	AB	4+	2+				
ADDB				CB	2	2	DB	4	2	FB	5	3	EB	4+	2+				
ADDD				C3	4	3	D3	6	2	F3	7	3	E3	6+	2+				
ANDA				84	2	2	94	4	2	B4	5	3	A4	4+	2+				
ANDB				C4	2	2	D4	4	2	F4	5	3	E4	4+	2+				
ANDCC				1C	3	2													
ASL							08	6	2	78	7	3	68	6+	2+				
ASLA	48	2	1																
ASLB	58	2	1																
ASR							07	6	2	77	7	3	67	6+	2+				
ASRA	47	2	1																
ASRB	57	2	1																
BCC																24	3	2	
BCS																25	3	2	
BEQ																27	3	2	
BGE																2C	3	2	
BGT																2E	3	2	
BHI																22	3	2	
BHS																24	3	2	
BITA				85	2	2	95	4	2	B5	5	3	A5	4+	2+				
BITB				C5	2	2	D5	4	2	F5	5	3	E5	4+	2+				

Continua

Modo de Endereçamento →	Inerente			Imediato			Direto			Extend			Indexado/ Indireto			Relativo			
Forma do Operando →				dado 8 ou 16 bits			end8			end16			ver Tab			Label ou desl.			
Mnemônico ↓	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Obs.
BLE																2F	3	2	
BLO																25	3	2	
BLS																23	3	2	
BLT																2D	3	2	
BMI																2B	3	2	
BNE																26	3	2	
BPL																2A	3	2	
BRA																20	3	2	
BRN																21	3	2	
BSR																8D	7	2	
BVC																28	3	2	
BVS																29	3	2	
CLR							0F	6	2	7F	7	3	6F	6+	2+				
CLRA	4F	2	1																
CLRB	5F	2	1																
CMPA				81	2	2	91	4	2	B1	5	3	A1	4+	2+				
CMPB				C1	2	2	D1	4	2	F1	5	3	E1	4+	2+				
CMPD				10 83	5	4	10 93	7	3	10 B3	8	4	10 A3	7+	3+				
CMPS				11 8C	5	4	11 9C	7	3	11 BC	8	4	11 AC	7+	3+				
CMPU				11 83	5	4	11 93	7	3	11 B3	8	4	11 A3	7+	3+				
CMPX				8C	4	3	9C	6	2	BC	7	3	AC	6+	2+				
CMPY				10 8C	5	4	10 9C	7	3	10 BC	8	4	10 AC	7+	3+				
COM							03	6	2	73	7	3	63	6+	2+				
COMA	43	2	1																
COMB	53	2	1																
CWAI				3C	20	2													

DAA	19	2	1																					
DEC							0A	6	2	7A	7	3	6A	6+	2+									
DECA	4A	2	1																					
DECB	5A	2	1																					
EORA				88	2	2	98	4	2	B8	5	3	A8	4+	2+									
EORB				C8	2	2	D8	4	2	F8	5	3	E8	4+	2+									
EXG				1E	8	2																		
INC							0C	6	2	7C	7	3	6C	6+	2+									2
INCA	4C	2	1																					
INCB	5C	2	1																					
JMP							0E	3	2	7E	4	3	6E	3+	2+									
JSR							9D	7	2	BD	8	3	AD	7+	2+									
LBCC																10 24		5(6)		4		1		
LBCS																10 25		5(6)		4		1		
LBEQ																10 27		5(6)		4		1		
LBGE																10 2C		5(6)		4		1		
LBGT																10 2E		5(6)		4		1		
LBHT																10 22		5(6)		4		1		
LBH5																10 24		5(6)		4		1		
LBLE																10 2F		5(6)		4		1		
LBLO																10 25		5(6)		4		1		
LBLS																10 23		5(6)		4		1		
LBLT																10 2D		5(6)		4		1		
LBM																10 2B		5(6)		4		1		
LBNE																10 26		5(6)		4		1		
LBPL																10 2A		5(6)		4		1		
LBRA																16		5		3				
LBRN																10 21		5		4				
LBSR																17		9		3				
LBVC																10 28		5(6)		4		1		
LBVS																10 29		5(6)		4		1		
LDA				86	2	2	96	4	2	B6	5	3	A6	4+	2+									
LDB				C6	2	2	D6	4	2	F6	5	3	E6	4+	2+									
LDD				CC	3	3	DC	5	2	FC	6	3	EC	5+	2+									

Continua

Modo de Endereçamento →	Inerente			Imediato			Direto			Extend			Indexado/ Indireto			Relativo			
Forma do Operando →				dado 8 ou 16 bits			end8			end16			ver Tab			Label ou desl.			
Mnemônico ↓	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Código	Ciclos	Nº bytes	Obs.
LDS				10 CE	4	4	10 DE	6	3	10 FE	7	4	10 EE	6+	3+				
LDU				CE	3	3	DE	5	2	FE	6	3	EE	5+	2+				
LDX				8E	3	3	9E	5	2	BE	6	3	AE	5+	2+				
LDY				10 8E	4	4	10 9E	6	3	10 BE	7	4	10 AE	6+	3+				
LEAS													32	4+	2+				
LEAU													33	4+	2+				
LEAX													30	4+	2+				
LEAY													31	4+	2+				
LSL							08	6	2	78	7	3	68	6+	2+				
LSLA	48	2	1																
LSLB	58	2	1																
LSR							04	6	2	74	7	3	64	6+	2+				
LSRA	44	2	1																
LSRB	54	2	1																
MUL	3D	11	1																
NEG							00	6	2	70	7	3	60	6+	2+				
NEGA	40	2	1																
NEGB	50	2	1																
NOP	12	2	1																
ORA				8A	2	2	9A	4	2	BA	5	3	AA	4+	2+				
ORB				CA	2	2	DA	4	2	FA	5	3	EA	4+	2+				
ORCC				1A	3	2													
PSHS				34	5+	2													2, 3
PSHU				36	5+	2													2, 3
PULS				35	5+	2													2, 3
PULU				37	5+	2													2, 3

ROL							09	6	2	79	7	3	69	6+	2+					
ROLA	49	2	1																	
ROLB	59	2	1																	
ROR							06	6	2	76	7	3	66	6+	2+					
RORA	46	2	1																	
RORB	56	2	1																	
RTI	3B	6/15	1																	
RTS	39	5	1																	
SBCA				82	2	2	92	4	2	B2	5	3	A2	4+	2+					
SBCB				C2	2	2	D2	4	2	F2	5	3	E2	4+	2+					
SEX	1D	2	1																	
STA							97	4	2	B7	5	3	A7	4+	2+					
STB							D7	4	2	F7	5	3	E7	4+	2+					
STD							DD	5	2	FD	6	3	ED	5+	2+					
STS							10 DF	6	3	10 FF	7	4	10 EF	6+	3+					
STU							DF	5	2	FF	6	3	EF	5+	2+					
STX							9F	5	2	BF	6	3	AF	5+	2+					
STY							10 9F	6	3	10 BF	7	4	10 AF	6+	3+					
SUBA				80	2	2	90	4	2	B0	5	3	A0	4+	2+					
SUBB				C0	2	2	D0	4	2	F0	5	3	E0	4+	2+					
SUBD				83	4	3	93	6	2	B3	7	3	A3	6+	2+					
SWI	3F	19	1																	
SWI2	10 3F	20	2																	
SWI3	11 3F	20	2																	
SYNC	13	2	1																	
TFR				1F	7	2														2
TST							0D	6	2	7D	7	3	6D	6+	2+					
TSTA	4D	2	1																	
TSTB	5D	2	1																	

4. MODOS INDIRETO E INDEXADO

Tipo	Forma	Não Indireto		+C	+B	Indireto		+C	+B
		Formato	Código			Formato	Código		
Deslocamento constante de r	Sem desl.	,r	pós-byte 1rr00100	0	0	[,r]	pós-byte 1rr10100	3	0
	desl. 5 bits	n,r	0rrnnnnn	1	0		Fica c/8 bits		
	desl. 8 bits	nn,r	1rr01000	1	1	[nn,r]	1rr11000	4	1
	desl. 16 bits	mmnn,r	1rr01001	4	2	[mmnn],r	1rr11001	7	1
Deslocamento de acumulador	desl. de A	A,r	1rr00110	1	0	[A,r]	1rr10110	4	0
	desl. de B	B,r	1rr00101	1	0	[B,r]	1rr10101	4	0
	desl. de D	D,r	1rr01011	4	0	[D,r]	1rr11011	7	0
Auto decrem./ increm. de r	pós-incr. 1	,r+	1rr00000	2	0		inviável		
	pós-incr. 2	,r++	1rr00001	3	0	[,r++]	1rr10001	6	0
	pós-decr. 1	,–r	1rr00010	2	0		inviável		
	pós-decr. 2	,–r	1rr00011	3	0	[,–r]	1rr10011	6	0
Deslocamento constante de PC	desl. 8 bits	rót,PCR	1xx01100	1	1	[rót,PCR]	1xx11100	4	1
	desl. 16 bits	rót,PCR	1xx01101	5	2	[rót,PCR]	1xx11101	8	2
Estendido indireto	end. 16 bits					[mmnn]	10011111	5	2

Observações:

1 – Os ciclos de clock entre parênteses se aplicam quando houver desvio.

2 – O operando é sempre um registrador.

3 – Estas instruções requerem 5 ciclos e mais 1 para cada byte manipulado.

5. INSTRUÇÕES

ABX	Soma B com X, resultado em X. Sem sinal
ADC	Soma memória com carry com acumulador A ou B. Resultado em A ou B
ADD	Soma memória com acumulador A, B ou D. Resultado em A, B ou D
AND	AND lógico entre A, B ou CC com operando
ASL	Desloca acumulador ou memória para a esquerda
ASR	Desloca acumulador ou memória para a direita
BCC	Desvio se carry igual a zero. Se salto > 1 byte use LBCC
BCS	Desvio se carry igual a 1. Se salto > 1 byte use LBCS
BEQ	Desvio se igual. Se salto > 1 byte use LBEQ
BGE	Desvio se maior ou igual (com sinal). Se salto > 1 byte use LBGE
BGT	Desvio se maior (com sinal). Se salto > 1 byte use LGBT
BHI	Desvio se maior (sem sinal). Se salto > 1 byte use LBHI
BHS	Desvio se maior ou igual (sem sinal). Se salto > 1 byte use LBHS
BIT	Testa bit do acumulador com o operando
BLE	Desvio se menor ou igual (com sinal). Se salto > 1 byte use LBLE
BLO	Desvio se menor (sem sinal). Se salto > 1 byte use LBLO
BLS	Desvio se menor ou igual (sem sinal). Se salto > 1 byte use LBLS
BLT	Desvio se menor (com sinal). Se salto > 1 byte use LBLT
BMI	Desvio se negativo. Se salto > 1 byte use LBMI
BNE	Desvio se diferente. Se salto > 1 byte use LBNE
BPL	Desvio se positivo. Se salto > 1 byte use LBPL
BRA	Desvio incondicional. Se salto > 1 byte use LBRA
BRN	Nunca desvia
BSR	Desvio para sub-rotina. Se salto > 1 byte use LBSR
BVC	Desvio se overflow igual a zero. Se salto > 1 byte use LBVC
BVS	Desvio se overflow igual a 1. Se salto > 1 byte use LBVS
CLR	Torna o registrador A ou B ou o lugar de memória zero
CMP	Compara memória com registrador
COM	Complementa acumulador ou memória
CWAI	Espera interrupção
DAA	Ajuste decimal do registrador A
DEC	Decrementa 1 do acumulador ou da memória
EOR	Exclusive OR entre memória e acumulador
EXG	Troca o conteúdo dos registradores
INC	Incrementa 1 ao acumulador ou à memória
JMP	Salto incondicional
JSR	Salto para a sub-rotina
LD	Carrega acumulador
LEA	Carrega endereço efetivo no registrador de 16 bits
LSL	Deslocamento lógico para a esquerda
LSR	Deslocamento lógico para a direita
MUL	Multiplica A por B, resultado em D (sem sinal)
NEG	Negação de acumulador ou memória
NOP	Nada faz

OR	OR lógico
PSH	Coloca o registrador no stack
PUL	Retira o registrador do stack
ROL	Rotação à esquerda
ROR	Rotação à direita
RTI	Retorna da interrupção
RTS	Retorna da sub-rotina
SBC	Subtrai memória e carry do acumulador
SEX	Coloca FF em A se B negativo; do contrário coloca 00 em A
ST	armazena o registrador na memória
SUB	Subtrai memória do registrador
SWI	Interrupção do software
SYNC	Sincronismo com evento externo
TFR	Transfere o conteúdo de um registrador para outro
TST	Testa acumulador ou memória

Impresso na
ERCA Editora e Gráfica Ltda.
Rua Silva Vale, 870 - Cavalcante
Rio de Janeiro - RJ

TRS COLOR

Este Guia de Referência contém todas as informações para você programar melhor o seu TRS-Color, tanto em BASIC quanto em linguagem de máquina. Apresenta farto material não coberto pelo manual do equipamento nem por nenhuma outra obra para este micro.

É dividido em seis partes, com os seguintes conteúdos:

- **MAPA DE MEMÓRIA** – principais endereços da ROM e suas finalidades
- **EDTASM** – comandos do Editor Assembly
- **BASIC** – comandos, mensagens de erro e técnicas de edição de programas
- **VÍDEO** – tabela de conjunto de cores, códigos de caracteres, modos gráficos, SAM e VDG
- **PEEK POKE EXEC** – entre outros, como dobrar e triplicar a velocidade do clock, desabilitar LIST, scroll lento e normal . . .
- **ASSEMBLY** – todas as instruções, técnicas de endereçamento e uso de registradores.

Esta obra é indispensável a todos que desejam extrair tudo que o TRS-Color pode dar.

ROBERTO VALOIS

é um ativo pesquisador do TRS-Color, sendo colaborador de diversas revistas especializadas da área.

ISBN 85-7001-366-3