Podprogramy MRS:

#ebdd	DEC > BIN	> HL=adresa raťazca 0-9 (iný znak ukončuje) < DE=16 bitova binárna hodnota < HL=adresa za posledným znakom retazca
#ebb8	HEX > BIN	> HL=adresa raťazca 0-f (iný znak ukončuje) < DE=16 bitova binárna hodnota < HL=adresa za posledným znakom retazca
#eb9d	BIN > HEX	> DE=16 bitova binárna hodnota > HL=bufer, kam sa uloží výsledný reťazec < HL=adresa za posledným znakom retazca
#eb75	BIN > DEC	> BC=16 bitova binárna hodnota > HL=bufer, kam sa uloží výsledný reťazec < HL=adresa za posledným znakom retazca
#d945	INKEY	< A=kód znaku + zvuková indikácia autorepeat
#d8e6	SCAN	< A=kód znaku
#d87b	CLEAR	> (#d88c)=hodnota atributu
#e01a	CLS	> (#dc8a)=hodnota BORDER+8 > (#d88c)=hodnota atributu
#dba2	OUTCHR	> A=kód znaku na obrazovku > (#d50c)=0-23 (#d50d)=0-63 pozícia
#dd77	PRINTCHR	> A=kód znaku na tlačiareň
#e017	CURSOR	> HL=pozícia kurzora H=0-23 L=0-63
#e008	OUTHEX	> HL=16 bitova binárna hodnota BIN > HEX & OUTCHR na obrazovku
#e00b	OUTDEC	> HL=16 bitova binárna hodnota BIN > DEC & OUTCHR na obrazovku
#e014	OUTBIN	> HL=16 bitova binárna hodnota
#d7e4	OUTBUF	> (#fe00)=textbuffer 64 znakov na obrazovku > H=0-23
#e7ce	CLSBUF	zmazanie textového buffera medzerami
#dadf	CHBLK	nájdenie konca textu v textbufferi < ZERO, 0=HL=addr konca, 1=HL=#fe40 same medzery
#dff5	OUTTXT	vypis textu ktory nasleduje za instrukciou CALL #DFF5 db #00 ukončuje vypis. Program vrati riadenie na addr po “db #00”

Bit	register F	1	0
7	S= SIGN bit7=0	JP M,nn	JP P,nn
6	Z= ZERO vysled. nulovy	JP Z,nn	JP NZ,nn
4	A= AUXILIARY H=HALF CARRY		
2	P= PARITY / OVERFLOW	JP PE,nn	JP PO,nn
1	N= ADD / SUBTRACT operacia neni odcita.		
0	C= CARRY	JP C,nn	JP NC,nn

DEBUGER:
X PCPC **instrukcia** SZAPC
AA BBCC DDEE HHLL XHXL YHYL SPSP

A	F	
B	C	
D	E	
H	L	
XH	IX	XL
YH	IY	YL
SP		
PC		
I	R	
A´	F´	
B´	C´	
D´	E´	
H´	L´	

AND B	Reg.A 11001100	204
	Reg.B 10001011	139
	Reg.A 10001000	136
OR B	Reg.A 11110000	240
	Reg.B 00111100	60
	Reg.A 11111100	252
XOR B	Reg.A 11110000	240
	Reg.B 00111100	60
	Reg.A 11001100	204
CPL	Reg.A 10111010	186
	Reg.A 01000101	136

DJNZ n
Reg.B je znizovany

IM0	NMI (CALL #0066)
IM1	RST #38 (KEYBOARD)
IM2	reg.I=XX CALL (#XX00-FF)
EI	povolí prerušenie
DI	zakáže prerušenie
HALT	čaká na prerušenie

CCF	C=invert
SCF	C=1
NEG	CPL+INC A

nairam
15.09.2000 nairam@nairam.sk

	RES	SET	BIT
BIT	Ø	1	TEST

RANDOMIZE USR 54885 (#D665)
HEX #ffff
BIN %00001111
;poznamka
;nizsi a vyssi bajt adresy
LD a,<addr ;a=#02 *addr-((addr/256)*256)*
LD a,>addr ;a=#80 *addr/256*
addr=#8002

navestie	instrukcia	text
EDIT	o stranu spat (23 riadkov)	
CAPS LOCK	o stranu dopredu (23 riadkov)	
TRUE VIDEO	listing o jedno pole dolava	
INV. VIDEO	listing o jedno pole doprava	
GRAPH	vytvori volnu poziciu	
DELETE	zmazanie znaku	
ENTER	vytvorenie prazdneho riadku	
SS+ENTER	skoci na zaciatok	
SS+SPACE	skoci na koniec	
CS+ENTER	skopirovanie riadku	
EXTEND	zmazanie riadku	
SS+Q	undo, ak kurzor neopustil riadok	
SS+W	oznacenie zaciatku bloku (cpb, dlb, rmc)	
SS+E	oznacenie konca bloku (cpb, dlb, rmc)	
BREAK	exit z EDITORA	
org 32768,49152	preklad na 49152 funguje po presunuti na orig. 32768	
ds 358	vytvorenie oblasti hodnot. ds 300,50,7,1	
dw #0102,#aabb	2-bajtove vyrazy, najprv nizsi a potom vyssi bajt	
db 65,65,65	hodnoty db ‘AAA’	
equ	alebo “=” priradenie hodnoty navestia	
end	koniec prekladu	
***f**	rychly mod prekladu bez vypisu	
***s**	vypina rychly mod prekladu	
***a**	vypisuje protokol o preklade	
***e**	vypisuje len chybne riadky	
***l**	vypisuje protokol na tlaciaren (#dd77)=add skoku, reg.A=znak	
***p**	nova strana na tlaciaren #0c, alebo konstanta pocet riadkov	
***t**	vypisuje protokol na obrazovke	
***cXXXX**	uklada vysledny kod od adresy XXXX	
***oAA,DD**	OUT (AA),DD vhodne na strankovanie ROM	
***z####**	oznacenie chyby pri preklade.	

MRS

R	nadstavenie 16-bit registra PC,AF,BC,DE,HL,IX,IY,SP
E	nadstavenie 8-bit registra A,F,B,C,D,E,H,L,R,I,X(HL),Y(HL)
I	bod prerusenia a pocet kol prebenutia aby sa zastavil (RST#10 a CHURCHL)
O	zmazanie bodu prerusenia
G	start programu
CS+G	start programu, ale program sa zastavi pri prvom prechode bodom prerusenia (I)
S	vykonanie jednej instrukcie s ochranou W
CS+S	vykonanie jednej instrukcie s ignorovanim ochran W
C	krokovanie s ochranou W
CS+C	krokovanie s ignorovanim ochran W
CS+X	preskocenie instrukcie
T	opakovany krokovaci rezim, L=zobrazi instruikciu
CS+T	opakovany krokovaci rezim s ignorovanim ochran W
N	zrychleny krokovaci rezim
CS+N	zrychleny krokovaci rezim s ignorovanim ochran W
W	ochrana M a P= Memory interval proti zapisu na adresy 1,2; PC interval 3,4
P	vypis casti pamete adresa,pocet riadkov (8 znakov)
M	modifikovanie pamete,EXTmode=povodny stav, TRUE a INVvideo=posun o 8 bajtov,SPACE=vloz ascii
CS+M	ako M s dalsou adresou
SS+M	ako M s aktualnou adresou
D	vypis od adresy dec,hex,8hex,ascii(32-127),ascii bez 128
CS+D	ako D s dalsou adresou
SS+D	ako D s aktualnou adresou
A	vypis od adresy dec,hex,instrukcia,hodnota hex,ascii
CS+A	ako A s dalsou adresou
SS+A	ako A s aktualnou adresou
X	vymena parovych registrov (EXX,EX AF,AF)
L	pocet stavovych informacii
Q	skok do dialogoveho rezimu
CS+Q	skok do BASICu
B	prepinanie znakovych sad ROM a 4*8
V	VAL vyraz pre editor
Z	skok do editora. BREAK=navrat
CS+Z	skok do editora. BREAK=preklad a navrat
SS+Z	preklad ASM
F	OUT port,bytes
CS+F	IN port
CS+K	SAVE KEY 1,2,3,4 na odkladanie PC adries
K	LOAD KEY 1,2,3,4 na odkladanie PC adries
Y	zakaz prerusenia (DI)
CS+Y	povolenie prerusenia (EI)
SS+Y	mod prerusenia 0,1,2
H	bezhlavickovy LOAD add, len, flag 0,1,2,3
SS+H	bezhlavickovy VERIFY add, len, flag 0,1,2,3
CS+H	bezhlavickove SAVE add, len, flag 0,1,2,3
J	LOAD “” CODE add,len
SS+J	VERIFY “” CODE add,len
CS+J	SAVE “” CODE add,len
SS+S	SPD 0=1500 baud 1=3000 baud 2=ROM
CS+U	RUN nnnn
SS+Q	SYS vypis MEMTOP, RAMTOP
SS+F	FILL zaplnenie bajtom add,len,bytes
SS+C	COPY oblasti pamete zdrojova add,len,cielova add
SS+V	COMP porovnanie pamete add,len,add2 S=vypis zhodov, D=vypis rozdielov,S=pozastavenie,Q=cont
SS+X	XOR vyxorovanie pamete add,len,bytes
SS+B	REVERSE obratenie sledu pamete add,len
CS+ENTER	HEA obsah hlavicky (0=basic,1=numbers,2=chars,3=bytes)
SS+SPACE	NMI call #0066
EXTEND	zapne dva stavove riadky
DELETE	CLS zmazanie obrazovky
GRAPH	CLS ale iba ATRIBUTY

0	1	#1	1	2	#2	2	4	#4	3	8	#8	4	16	#10	5	32	#20
6	64	#40	7	128	#80	8	256	#100	9	512	#200	10	1024	#400	11	2048	#800
12	4096	#1000	13	8192	#2000	14	16384	#4000	15	32768	#8000						

RLC A=A*2 — C ← 7 6 5 4 3 2 1 0 ← ; IF A>127 A=A(255-A)
RRC A=A/2 — → 7 6 5 4 3 2 1 0 → C ; IF BITO=1 A=A/2+128
RL A=A*2 — C ← 7 6 5 4 3 2 1 0 ← ; IF C=1 A=A-(255-A)
RR A=A/2 — → 7 6 5 4 3 2 1 0 → C ; IF C=1 A=A/2+128
SLA A=A*2 — C ← 7 6 5 4 3 2 1 0 ← Ø ; IF BIT7=1 A=A*2-128
SLL A=A*2+1 — C ← 7 6 5 4 3 2 1 0 ← 1 ; IF A>127 A=A-(255-A)
SRA — → 7 6 5 4 3 2 1 0 → C
SRL A=A/2 — Ø → 7 6 5 4 3 2 1 0 → C
RLD — (HL)=(HL)+A, A=A-2 — A | 7 4 3 0 | 7 4 3 0 | (HL)
RRD — (HL)=(HL)-A, A=A+2 — A | 7 4 3 0 | 7 4 3 0 | (HL)

ldd	LOAD zdrojaku
svd	SAVE zdrojaku
ldc	LOAD code
mrd	MERGE suboru
era	ERASE suboru
dev	DEVAST
cat,dir	vypis suborov
dos	skok do TR-DOS

+	-
INC	DEC
ADD	SUB
ADC	SBC
DAA	

LDD	(DE)<(HL), HL=HL-1, DE=DE-1, BC=BC-1; if BC=0, then p=0
LDI	(DE)<(HL), HL=HL+1, DE=DE+1, BC=BC-1; if BC=0, then p=0
LDDR	(DE)<(HL), HL=HL-1, DE=DE-1, BC=BC-1; ukončí ak BC=0
LDIR	(DE)<(HL), HL=HL+1, DE=DE+1, BC=BC-1; ukončí ak BC=0
CP n	platí porovnanie A==n?
CPD	CP (HL); HL=HL-1; BC=BC-1
CPI	CP (HL); HL=HL+1; BC=BC-1
CPDR	CP (HL); HL=HL-1; BC=BC-1; ukončí ak A=(HL) or BC=0
CPIR	CP (HL); HL=HL+1; BC=BC-1; ukončí ak A=(HL) or BC=0
IN reg,port	čítanie z portu
IND	IN (HL),(C); B=B-1; HL=HL-1
INI	IN (HL),(C); B=B-1; HL=HL+1
INDR	IN (HL),(C); B=B-1; HL=HL-1 ukončí ak B=0
INIR	IN (HL),(C); B=B-1; HL=HL+1 ukončí ak B=0

DIALOGOVY REZIM edi>

INI	vymazanie celeho zdrojaku
PRE	pretransformovanie zdrojakov starsich verzii
loa	LOAD tape
ver	VERIFY tape
mer	MERGE tape
sav	SAVE tape
spd	speed tape save 0=1500 baud 1=2400 2=ROM
ln=	skok na riadok
ln-	skok na zaciatok
ln+	skok na koniec
cpb	copy od M do M na kurzor “.”=SS+W az SS+E v editore
dlb	mazanie od M do N “.”=SS+W az SS+E v editore
rmc;	zmazanie komentarov za “;” “.”=SS+W az SS+E v editore
rmc*	zmazanie komentarov v bloku text
rmc!	zmazanie vsetkych komentarovinsdis s navestiami
dis	preklad z adresy do editora na instrukcie
dis*	db #nn,#nn,#nn,#nn
dis:	dw #nnnn,#nnnn,#nnnn,#nnnn
dis$	db ‘a’,’a’,’a’
dis!	datove struktury, rychly preklad asm
asm	asembleruj zdrojak (printer(#dd77)=add skoku, reg.A=znak)
ald	asm + dbg
dbg	debuger (ENTER)
mon	exit to BASIC
run	USR xx
new	nadstavenie RAMTOP
cls	vycisti obrazovku
val	vyraz bin,hex,dec
sys	vypise memtop(zdrojovy text), ramtop
let navestie vyraz	priradi navestiu hodnotu. ASM rusi
ref	vyhlada retazec
alt	zamena navesti 1. za 2.
mod	0=hex 1=dec
hea	obsah hlavicky (0=basic,1=numbers,2=chars,3=bytes)