

STM8 ASSEMBLER COMANDS. Page1.

LD A,MR LD A,M	LD A,MN LD A,M	LD A,(Ri) LD A,@Ri	LD A,(MR,Ri) LD A,@(RM+Ri)	LD A,(MN,Ri) LD A,@(RM+Ri)	LD A,(MR,SP) LD A,@(RM+SP)	LD A,[MR.w] LD A,@M	LD A,[MN.w] LD A,@M	LD A,([MR.w],Ri) LD A,@(M+X)	LD A,([MN.w],X) LD A,@(M+X)
LD MR,A LD M,A	LD MN,A LD M,A	LD (Ri),A LD @Ri,A	LD (MR,Ri),A LD @((RM+Ri),A	LD (MN,Ri),A LD @((RM+Ri),A	LD (MR,SP),A LD @((RM+SP),A	LD A,[MR.w] LD @M,A	LD [MN.w],A LD @M,A	LD ([MR.w],Ri),A LD @((M+X),A	LD ([MN.w],X),A LD @((M+X),A

LD A,RiL LD A,RiL	LD A,RiH LD A,RiH
LD RiL,A LD RiL,A	LD RiH,A LD RiH,A

LDW Ri,MR LDW Ri_a,@(RM+Ri_a)	LDW Ri,MN LDW Ri_a,@(RM+Ri_a)	LDW Ri_a,(Ri_a) LDW Ri_a,@(RM+Ri_a)	LDW Ri_a,(MR,Ri_a) LDW Ri_a,@(RM+Ri_a)	LDW Ri_a,(MN,Ri_a) LDW Ri_a,@(RM+Ri_a)	LDW Ri,(MR,SP) LDW Ri,@(RM+SP)	LDW Ri,[MR.w] LDW Ri,@M	LDW Ri_a,([MR.w],Ri_a) LDW Ri_a,@(M+Ri_a)	LDW Ri_a,([MN.w],Ri_a) LDW Ri_a,@(M+Ri_a)
LDW MR,Ri LDW @((RM+Ri_a),Ri_b)	LDW MN,Ri LDW @((RM+Ri_a),Ri_b)	LDW (Ri_a),Ri_b LDW @((RM+Ri_a),Ri_b)	LDW (MR,Ri_a),Ri_b LDW @((RM+Ri_a),Ri_b)	LDW (MN,Ri_a),Ri_b LDW @((RM+Ri_a),Ri_b)	LDW (MR,SP),Ri LDW @((RM+SP),Ri)	LDW [MR.w],Ri LDW @M,Ri	LDW ([MR.w],Ri_a),Ri_b LDW @((M+Ri_a),Ri_b)	LDW ([MN.w],Ri_a),Ri_b LDW @((M+Ri_a),Ri_b)

BSET <i>M0,#bit</i> SETB M.#bit	BRES <i>M0,#bit</i> CLR M.#bit	BCPL <i>M0,#bit</i> CPL M.#bit				
BCCM <i>M0,#bit</i> MOV M.#bit,CY	BTJF <i>M0,#bit,rel</i> MOV CY,M.#bit JNC rel	BTJT <i>M0,#bit,rel</i> MOV CY,M.#bit JC rel	SCF SETB CY	RCF CLR CY	CCF CPL CY	RVF CLR V

LDF A,MF LDF A,@(MM+Ri)	LDF A,(MF,Ri) LDF A,@M	LDF A,[MN.e] LDF A,@M	LDF A,([MN.e],Ri) LDF A,@(M+Ri)
LDF MF,A LDF @((MM+Ri),A)	LDF (MF,Ri),A LDF @M,A	LDF [MN.e],A LDF @M,A	LDF ([MN.e],Ri),A LDF @((M+Ri),A)

MOV MR,MR MOV MR,MR	LDW X,Y LDW X,Y	LDW Ri,SP LDW Ri,SP	EXG A,XL EXG A,XL	EXG A,M0 EXG A,M0
MOV M0,M0 MOV M0,M0	LDW Y,X LDW Y,X	LDW SP,Ri LDW SP,Ri	EXG A,YL EXG A,YL	EXGW X,Y EXGW X,Y

PUSH #d8 PUSH #d8	PUSH A PUSH A	PUSH CC PUSH CC	PUSH M0 PUSH M0	PUSHW Ri PUSHW Ri
	POP A POP A	POP CC POP CC	POP M0 POP M0	POPW Ri POPW Ri

LD A,#d8 LD A,#d8	LDW Ri,#d16 LDW Ri,#d16	MOV M0,#d8 MOV M0,#d8
CLR A CLR A	CLR MR CLR MR	CLR M0 CLR M0

CPL A CPL @Ri	CPL MR CPL @((MM+Ri)	CPL M0 CPL @((MM+Ri)	CPL (Ri) CPL @Ri	CPL (MR,Ri) CPL @((MM+Ri)	CPL (M0,Ri) CPL @((MM+Ri)	CPL (MR,SP) CPL @((MM+SP)	CPL [MR.w] CPL @M	CPL [M0.w] CPL @M	CPL ([MR.w],Ri) CPL @((M+Ri)	CPL ([M0.w],X) CPL @((M+Ri)	CLRW Ri CLR @Ri
NEG A NEG @Ri	NEG MR NEG @((MM+Ri)	NEG M0 NEG @((MM+Ri)	NEG (Ri) NEG @Ri	NEG (MR,Ri) NEG @((MM+Ri)	NEG (M0,Ri) NEG @((MM+Ri)	NEG (MR,SP) NEG @((MM+SP)	NEG [MR.w] NEG @M	NEG [M0.w] NEG @M	NEG ([MR.w],Ri) NEG @((M+Ri)	NEG ([M0.w],X) NEG @((M+Ri)	NEGW Ri NEG @Ri
INC A INC @Ri	INC MR INC @((MM+Ri)	INC M0 INC @((MM+Ri)	INC (Ri) INC @Ri	INC (MR,Ri) INC @((MM+Ri)	INC (M0,Ri) INC @((MM+Ri)	INC (MR,SP) INC @((MM+SP)	INC [MR.w] INC @M	INC [M0.w] INC @M	INC ([MR.w],Ri) INC @((M+Ri)	INC ([M0.w],X) INC @((M+Ri)	INCW Ri INC @Ri
DEC A DEC @Ri	DEC MR DEC @((MM+Ri)	DEC M0 DEC @((MM+Ri)	DEC (Ri) DEC @Ri	DEC (MR,Ri) DEC @((MM+Ri)	DEC (M0,Ri) DEC @((MM+Ri)	DEC (MR,SP) DEC @((MM+SP)	DEC [MR.w] DEC @M	DEC [M0.w] DEC @M	DEC ([MR.w],Ri) DEC @((M+Ri)	DEC ([M0.w],X) DEC @((M+Ri)	DECW Ri DEC @Ri
SWAP A SWAP @Ri	SWAP MR SWAP @((MM+Ri)	SWAP M0 SWAP @((MM+Ri)	SWAP (Ri) SWAP @Ri	SWAP (MR,Ri) SWAP @((MM+Ri)	SWAP (M0,Ri) SWAP @((MM+Ri)	SWAP (MR,SP) SWAP @((MM+SP)	SWAP [MR.w] SWAP @M	SWAP [M0.w] SWAP @M	SWAP ([MR.w],Ri) SWAP @((M+Ri)	SWAP ([M0.w],X) SWAP @((M+Ri)	SWAPW Ri SWAP @Ri
TNZ A TST @Ri	TNZ MR TST @((MM+Ri)	TNZ M0 TST @((MM+Ri)	TNZ (Ri) TST @Ri	TNZ (MR,Ri) TST @((MM+Ri)	TNZ (M0,Ri) TST @((MM+Ri)	TNZ (MR,SP) TST @((MM+SP)	TNZ [MR.w] TST @M	TNZ [M0.w] TST @M	TNZ ([MR.w],Ri) TST @((M+Ri)	TNZ ([M0.w],X) TST @((M+Ri)	TNZW Ri TST @Ri

NOP	WFE SETB IT CLR IO	WFI SETB IT CLR IO	BREAK	HALT SETB IT CLR IO
------------	---	---	--------------	--

#bit - номер бита 0-7
#d8 - непосредственные данные 0-0xFF
#d16 - непосредственные данные 0-0xFFFF
rel - короткий переход +127/-128 относительно текущего значения PC:PCH:PCL
MR диапазон адресов 0x000000-0x0000FF
M0 диапазон адресов 0x000000-0x00FFFF частный случай от MN
MN диапазон адресов 0xnn0000-0xnnFFFF
MF диапазон адресов 0x000000-0xFFFFF
В качестве MR,M0,MN,MF может использоваться прямоуказанный адрес или имя метки определенное первой ячейке данных
Ri - индексные регистры X или Y
Ri_a,(Ri_a)=X,(X) или Y,(Y)
Ri_a,(Ri_b)=X,(Y) или Y,(X)

JRA rel JMP #MN	JP MN JMP @MN	JP (Ri) JMP @Ri	JP (MR,Ri) JMP @((MM+Ri)	JP (MN,Ri) JMP @((MM+Ri)	JP [MR.w] JMP @MR	JP [MN.w] JMP @MN	JP ([MR.w],Ri) JMP @((MM+Ri)	JP ([MN.w],X) JMP @((MN+X)	JPF MF LJMP #MF	JPF [MN.e] LJMP @MN
CALLR rel CALL #MN	CALL MN CALL @MN	CALL (Ri) ICALL @Ri	CALL (MR,Ri) ICALL @((MM+Ri)	CALL (MN,Ri) ICALL @((MM+Ri)	CALL [MR.w] ICALL @MR	CALL [MN.w] ICALL @MN	CALL ([MR.w],Ri) ICALL @((MM+Ri)	CALL ([MN.w],X) ICALL @((MN+X)	CALLF MF LCALL #MF	CALLF [MN.e] LCALL @MN

RET POP PCH POP PCL	TRAP PUSH PCL PUSH PCH PUSH PCE PUSH YL PUSH YH PUSH XH PUSH A PUSH CC PCE:PCH:PCL=TRAP:INC SETB IT SETB IO	IRET POP CC POP A POP XH POP XL POP YH POP YL POP PCE POP PCH POP PCL
RETF POP PCE POP PCH POP PCL		

суффиксы длины данных по указанному адресу
.b - байт
.w - 2 байта (слово)
.e -3 байта (extended адрес)
.I -4 байта (в основном для директив DC. или DS.)

TNZ=AND изменяет только N и Z флаги по результату