

STM8 ASSEMBLER COMANDS. Page1.

LD <i>A, #d8</i>	MOV <i>M0, #d8</i>	MOV <i>MR, MR</i>	LD <i>A, RiL</i>	LD <i>A, RiH</i>	LDW <i>X, Y</i>	LDW <i>Ri, SP</i>	EXG <i>A, XL</i>	EXG <i>A, M0</i>	PUSH <i>A</i>	PUSH <i>CC</i>	PUSH <i>M0</i>	PUSHW <i>Ri</i>
LDW <i>Ri, #d16</i>	PUSH <i>#d8</i>	MOV <i>M0, M0</i>	LD <i>RiL, A</i>	LD <i>RiH, A</i>	LDW <i>Y, X</i>	LDW <i>SP, Ri</i>	EXG <i>A, YL</i>	EXGW <i>X, Y</i>	POP <i>A</i>	POP <i>CC</i>	POP <i>M0</i>	POPW <i>Ri</i>

LD <i>A, MR</i>	LD <i>A, MN</i>	LD <i>A, (Ri)</i>	LD <i>A, (MR, Ri)</i>	LD <i>A, (MN, Ri)</i>	LD <i>A, (MR, SP)</i>	LD <i>A, [MR.w]</i>	LD <i>A, [MN.w]</i>	LD <i>A, ([MR.w], Ri)</i>	LD <i>A, ([MN.w], X)</i>	LDF <i>A, MF</i>	LDF <i>A, (MF, Ri)</i>	LDF <i>A, [MN.e]</i>	LDF <i>A, ([MN.e], Ri)</i>
<i>LD A, M</i>	<i>LD A, M</i>	<i>LD A, @Ri</i>	<i>LD A, @(RM+Ri)</i>	<i>LD A, @(RM+Ri)</i>	<i>LD A, @(RM+SP)</i>	<i>LD A, @M</i>	<i>LD A, @M</i>	<i>LD A, @(M+X)</i>	<i>LD A, @(M+X)</i>	<i>LDF A, @MF</i>	<i>LDF A, @(MF+Ri)</i>	<i>LDF A, @M</i>	<i>LDF A, @(M+Ri)</i>
LD <i>MR, A</i>	LD <i>MN, A</i>	LD <i>(Ri), A</i>	LD <i>(MR, Ri), A</i>	LD <i>(MN, Ri), A</i>	LD <i>(MR, SP), A</i>	LD <i>A, [MR.w]</i>	LD <i>[MN.w], A</i>	LD <i>([MR.w], Ri), A</i>	LD <i>([MN.w], X), A</i>	LDF <i>MF, A</i>	LDF <i>(MF, Ri), A</i>	LDF <i>[MN.e], A</i>	LDF <i>([MN.e], Ri), A</i>
<i>LD M, A</i>	<i>LD M, A</i>	<i>LD @Ri, A</i>	<i>LD @(RM+Ri), A</i>	<i>LD @(RM+Ri), A</i>	<i>LD @(RM+SP), A</i>	<i>LD @M, A</i>	<i>LD @M, A</i>	<i>LD @(M+X), A</i>	<i>LD @(M+X), A</i>	<i>LDF @MF, A</i>	<i>LDF @(MF+Ri), A</i>	<i>LDF @M, A</i>	<i>LDF @(M+Ri), A</i>

AND <i>A, #d8</i>	AND <i>A, MR</i>	AND <i>A, MN</i>	AND <i>A, (Ri)</i>	AND <i>A, (MR, Ri)</i>	AND <i>A, (MN, Ri)</i>	AND <i>A, (MR, SP)</i>	AND <i>A, [MR.w]</i>	AND <i>A, [MN.w]</i>	AND <i>A, ([MR.w], Ri)</i>	AND <i>A, ([MN.w], X)</i>
<i>AND A, #d8</i>	<i>AND A, M</i>	<i>AND A, M</i>	<i>AND A, @Ri</i>	<i>AND A, @(RM+Ri)</i>	<i>AND A, @(RM+Ri)</i>	<i>AND A, @(RM+SP)</i>	<i>AND A, @M</i>	<i>AND A, @M</i>	<i>AND A, @(M+Ri)</i>	<i>AND A, @(M+X)</i>
OR <i>A, #d8</i>	OR <i>A, MR</i>	OR <i>A, MN</i>	OR <i>A, (Ri)</i>	OR <i>A, (MR, Ri)</i>	OR <i>A, (MN, Ri)</i>	OR <i>A, (MR, SP)</i>	OR <i>A, [MR.w]</i>	OR <i>A, [MN.w]</i>	OR <i>A, ([MR.w], Ri)</i>	OR <i>A, ([MN.w], X)</i>
<i>OR A, #d8</i>	<i>OR A, M</i>	<i>OR A, M</i>	<i>OR A, @Ri</i>	<i>OR A, @(RM+Ri)</i>	<i>OR A, @(RM+Ri)</i>	<i>OR A, @(RM+SP)</i>	<i>OR A, @M</i>	<i>OR A, @M</i>	<i>OR A, @(M+Ri)</i>	<i>OR A, @(M+X)</i>
XOR <i>A, #d8</i>	XOR <i>A, MR</i>	XOR <i>A, MN</i>	XOR <i>A, (Ri)</i>	XOR <i>A, (MR, Ri)</i>	XOR <i>A, (MN, Ri)</i>	XOR <i>A, (MR, SP)</i>	XOR <i>A, [MR.w]</i>	XOR <i>A, [MN.w]</i>	XOR <i>A, ([MR.w], Ri)</i>	XOR <i>A, ([MN.w], X)</i>
<i>XOR A, #d8</i>	<i>XOR A, M</i>	<i>XOR A, M</i>	<i>XOR A, @Ri</i>	<i>XOR A, @(RM+Ri)</i>	<i>XOR A, @(RM+Ri)</i>	<i>XOR A, @(RM+SP)</i>	<i>XOR A, @M</i>	<i>XOR A, @M</i>	<i>XOR A, @(M+Ri)</i>	<i>XOR A, @(M+X)</i>
CP <i>A, #d8</i>	CP <i>A, MR</i>	CP <i>A, MN</i>	CP <i>A, (Ri)</i>	CP <i>A, (MR, Ri)</i>	CP <i>A, (MN, Ri)</i>	CP <i>A, (MR, SP)</i>	CP <i>A, [MR.w]</i>	CP <i>A, [MN.w]</i>	CP <i>A, ([MR.w], Ri)</i>	CP <i>A, ([MN.w], X)</i>
<i>CP A, #d8</i>	<i>CP A, M</i>	<i>CP A, M</i>	<i>CP A, @Ri</i>	<i>CP A, @(RM+Ri)</i>	<i>CP A, @(RM+Ri)</i>	<i>CP A, @(RM+SP)</i>	<i>CP A, @M</i>	<i>CP A, @M</i>	<i>CP A, @(M+Ri)</i>	<i>CP A, @(M+X)</i>
BCP <i>A, #d8</i>	BCP <i>A, MR</i>	BCP <i>A, MN</i>	BCP <i>A, (Ri)</i>	BCP <i>A, (MR, Ri)</i>	BCP <i>A, (MN, Ri)</i>	BCP <i>A, (MR, SP)</i>	BCP <i>A, [MR.w]</i>	BCP <i>A, [MN.w]</i>	BCP <i>A, ([MR.w], Ri)</i>	BCP <i>A, ([MN.w], X)</i>
<i>BCP A, #d8</i>	<i>BCP A, M</i>	<i>BCP A, M</i>	<i>BCP A, @Ri</i>	<i>BCP A, @(RM+Ri)</i>	<i>BCP A, @(RM+Ri)</i>	<i>BCP A, @(RM+SP)</i>	<i>BCP A, @M</i>	<i>BCP A, @M</i>	<i>BCP A, @(M+Ri)</i>	<i>BCP A, @(M+X)</i>
ADC <i>A, #d8</i>	ADC <i>A, MR</i>	ADC <i>A, MN</i>	ADC <i>A, (Ri)</i>	ADC <i>A, (MR, Ri)</i>	ADC <i>A, (MN, Ri)</i>	ADC <i>A, (MR, SP)</i>	ADC <i>A, [MR.w]</i>	ADC <i>A, [MN.w]</i>	ADC <i>A, ([MR.w], Ri)</i>	ADC <i>A, ([MN.w], X)</i>
<i>ADC A, #d8</i>	<i>ADC A, M</i>	<i>ADC A, M</i>	<i>ADC A, @Ri</i>	<i>ADC A, @(RM+Ri)</i>	<i>ADC A, @(RM+Ri)</i>	<i>ADC A, @(RM+SP)</i>	<i>ADC A, @M</i>	<i>ADC A, @M</i>	<i>ADC A, @(M+Ri)</i>	<i>ADC A, @(M+X)</i>
ADD <i>A, #d8</i>	ADD <i>A, MR</i>	ADD <i>A, MN</i>	ADD <i>A, (Ri)</i>	ADD <i>A, (MR, Ri)</i>	ADD <i>A, (MN, Ri)</i>	ADD <i>A, (MR, SP)</i>	ADD <i>A, [MR.w]</i>	ADD <i>A, [MN.w]</i>	ADD <i>A, ([MR.w], Ri)</i>	ADD <i>A, ([MN.w], X)</i>
<i>ADD A, #d8</i>	<i>ADD A, M</i>	<i>ADD A, M</i>	<i>ADD A, @Ri</i>	<i>ADD A, @(RM+Ri)</i>	<i>ADD A, @(RM+Ri)</i>	<i>ADD A, @(RM+SP)</i>	<i>ADD A, @M</i>	<i>ADD A, @M</i>	<i>ADD A, @(M+Ri)</i>	<i>ADD A, @(M+X)</i>
SBC <i>A, #d8</i>	SBC <i>A, MR</i>	SBC <i>A, MN</i>	SBC <i>A, (Ri)</i>	SBC <i>A, (MR, Ri)</i>	SBC <i>A, (MN, Ri)</i>	SBC <i>A, (MR, SP)</i>	SBC <i>A, [MR.w]</i>	SBC <i>A, [MN.w]</i>	SBC <i>A, ([MR.w], Ri)</i>	SBC <i>A, ([MN.w], X)</i>
<i>SBC A, #d8</i>	<i>SBC A, M</i>	<i>SBC A, M</i>	<i>SBC A, @Ri</i>	<i>SBC A, @(RM+Ri)</i>	<i>SBC A, @(RM+Ri)</i>	<i>SBC A, @(RM+SP)</i>	<i>SBC A, @M</i>	<i>SBC A, @M</i>	<i>SBC A, @(M+Ri)</i>	<i>SBC A, @(M+X)</i>
SUB <i>A, #d8</i>	SUB <i>A, MR</i>	SUB <i>A, MN</i>	SUB <i>A, (Ri)</i>	SUB <i>A, (MR, Ri)</i>	SUB <i>A, (MN, Ri)</i>	SUB <i>A, (MR, SP)</i>	SUB <i>A, [MR.w]</i>	SUB <i>A, [MN.w]</i>	SUB <i>A, ([MR.w], Ri)</i>	SUB <i>A, ([MN.w], X)</i>
<i>SUB A, #d8</i>	<i>SUB A, M</i>	<i>SUB A, M</i>	<i>SUB A, @Ri</i>	<i>SUB A, @(RM+Ri)</i>	<i>SUB A, @(RM+Ri)</i>	<i>SUB A, @(RM+SP)</i>	<i>SUB A, @M</i>	<i>SUB A, @M</i>	<i>SUB A, @(M+Ri)</i>	<i>SUB A, @(M+X)</i>

CPW <i>Ri, #d16</i>	CPW <i>Ri, MR</i>	CPW <i>Ri, MN</i>	CPW <i>Ri_a, (Ri_b)</i>	CPW <i>Ri_a, (MR, Ri_b)</i>	CPW <i>Ri_a, (MN, Ri_b)</i>	CPW <i>X, (MR, SP)</i>	CPW <i>Ri, [MR.w]</i>	CPW <i>X, [MN.w]</i>
<i>CPW Ri, #d16</i>	<i>CPW Ri, M</i>	<i>CPW Ri, M</i>	<i>CPW Ri_a, @Ri_b</i>	<i>CPW Ri_a, @(RM+Ri_b)</i>	<i>CPW Ri_a, @(RM+Ri_b)</i>	<i>CPW X, @(RM+SP)</i>	<i>CPW Ri, @MR</i>	<i>CPW X, @MN</i>

ADDW <i>SP, #d8</i>	ADDW <i>Ri, #d16</i>	ADDW <i>Ri, MR</i>	ADDW <i>Ri, MN</i>	ADDW <i>Ri, (MR, SP)</i>	SUBW <i>SP, #d8</i>	SUBW <i>Ri, #d16</i>	SUBW <i>Ri, MR</i>	SUBW <i>Ri, MN</i>	SUBW <i>Ri, (MR, SP)</i>	MUL <i>Ri, A</i>	DIV <i>Ri, A</i>	DIVW <i>X, Y</i>
<i>ADDW SP, #d8</i>	<i>ADDW Ri, #d16</i>	<i>ADDW Ri, M</i>	<i>ADDW Ri, M</i>	<i>ADDW Ri, @(RM+SP)</i>	<i>SUBW SP, #d8</i>	<i>SUBW Ri, #d16</i>	<i>SUBW Ri, M</i>	<i>SUBW Ri, M</i>	<i>SUBW Ri, @(RM+SP)</i>	<i>MUL Ri, LA resultat e Ri</i>	<i>DIV Ri, A целое e Ri, остаток e A</i>	<i>DIVW X, Y целое e X, остаток e Y</i>

NOP

WFE	WFI	BREAK	HALT
	<i>SETB I1 CLR I0</i>		<i>SETB I1 CLR I0</i>

#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_b)=X, (X) или Y, (Y)
Ri_a, (Ri_b)=X, (Y) или Y, (X)

(.....) - адрес ячейки памяти
[.....] - содержимое указателя по адресу M...
#M - адрес ячейки памяти
#M - содержимое ячейки памяти
суффиксы длины данных по указанному адресу
.b - байт
.w - 2 байта (слово)
.e -3 байта (extended адрес)
.l -4 байта (в основном для директив DC. или DS.)

TNZ=AND изменяет только N и Z флаги по результату