

SLL / SLA <i>A</i>	SLL / SLA <i>MR</i>	SLL / SLA <i>M0</i>	SLL / SLA <i>(Ri)</i> SLL/SLA @Ri	SLL / SLA <i>(MR,Ri)</i> SLL/SLA @(M+Ri)	SLL / SLA <i>(M0,Ri)</i> SLL/SLA @(M+Ri)	SLL / SLA <i>(MR,SP)</i> SLL/SLA @(M+SP)	SLL / SLA <i>[MR.w]</i> SLL/SLA @M	SLL / SLA <i>[M0.w]</i> SLL/SLA @M	SLL / SLA <i>((MR.w),Ri)</i> SLL/SLA @(M+Ri)	SLL / SLA <i>((M0.w),X)</i> SLL/SLA @(M+Ri)	SLLW / SLAW <i>Ri</i>
SRA <i>A</i>	SRA <i>MR</i>	SRA <i>M0</i>	SRA <i>(Ri)</i> SRA @Ri	SRA <i>(MR,Ri)</i> SRA @(M+Ri)	SRA <i>(M0,Ri)</i> SRA @(M+Ri)	SRA <i>(MR,SP)</i> SRA @(M+SP)	SRA <i>[MR.w]</i> SRA @M	SRA <i>[M0.w]</i> SRA @M	SRA <i>((MR.w),Ri)</i> SRA @(M+Ri)	SRA <i>((M0.w),X)</i> SRA @(M+Ri)	SRAW <i>Ri</i>
SRL <i>A</i>	SRL <i>MR</i>	SRL <i>M0</i>	SRL <i>(Ri)</i> SRL @Ri	SRL <i>(MR,Ri)</i> SRL @(M+Ri)	SRL <i>(M0,Ri)</i> SRL @(M+Ri)	SRL <i>(MR,SP)</i> SRL @(M+SP)	SRL <i>[MR.w]</i> SRL @M	SRL <i>[M0.w]</i> SRL @M	SRL <i>((MR.w),Ri)</i> SRL @(M+Ri)	SRL <i>((M0.w),X)</i> SRL @(M+Ri)	SRLW <i>Ri</i>
RLC <i>A</i>	RLC <i>MR</i>	RLC <i>M0</i>	RLC <i>(Ri)</i> RLC @Ri	RLC <i>(MR,Ri)</i> RLC @(M+Ri)	RLC <i>(M0,Ri)</i> RLC @(M+Ri)	RLC <i>(MR,SP)</i> RLC @(M+SP)	RLC <i>[MR.w]</i> RLC @M	RLC <i>[M0.w]</i> RLC @M	RLC <i>((MR.w),Ri)</i> RLC @(M+Ri)	RLC <i>((M0.w),X)</i> RLC @(M+Ri)	RLCW <i>Ri</i>
RRC <i>A</i>	RRC <i>MR</i>	RRC <i>M0</i>	RRC <i>(Ri)</i> RRC @Ri	RRC <i>(MR,Ri)</i> RRC @(M+Ri)	RRC <i>(M0,Ri)</i> RRC @(M+Ri)	RRC <i>(MR,SP)</i> RRC @(M+SP)	RRC <i>[MR.w]</i> RRC @M	RRC <i>[M0.w]</i> RRC @M	RRC <i>((MR.w),Ri)</i> RRC @(M+Ri)	RRC <i>((M0.w),X)</i> RRC @(M+Ri)	RRCW <i>Ri</i>

AND <i>A,#d8</i> AND A,#d8	AND <i>A,MR</i> AND A,M	AND <i>A,MN</i> AND A,M	AND <i>A,(Ri)</i> AND A,@Ri	AND <i>A,(MR,Ri)</i> AND A,@(M+Ri)	AND <i>A,(MN,Ri)</i> AND A,@(M+Ri)	AND <i>A,(MR,SP)</i> AND A,@(M+SP)	AND <i>A,[MR.w]</i> AND A,@M	AND <i>A,[MN.w]</i> AND A,@M	AND <i>A,((MR.w),Ri)</i> AND A,@(M+Ri)	AND <i>A,((MN.w),X)</i> AND A,@(M+X)
OR <i>A,#d8</i> OR A,#d8	OR <i>A,MR</i> OR A,M	OR <i>A,MN</i> OR A,M	OR <i>A,(Ri)</i> OR A,@Ri	OR <i>A,(MR,Ri)</i> OR A,@(M+Ri)	OR <i>A,(MN,Ri)</i> OR A,@(M+Ri)	OR <i>A,(MR,SP)</i> OR A,@(M+SP)	OR <i>A,[MR.w]</i> OR A,@M	OR <i>A,[MN.w]</i> OR A,@M	OR <i>A,((MR.w),Ri)</i> OR A,@(M+Ri)	OR <i>A,((MN.w),X)</i> OR A,@(M+X)
XOR <i>A,#d8</i> XOR A,#d8	XOR <i>A,MR</i> XOR A,M	XOR <i>A,MN</i> XOR A,M	XOR <i>A,(Ri)</i> XOR A,@Ri	XOR <i>A,(MR,Ri)</i> XOR A,@(M+Ri)	XOR <i>A,(MN,Ri)</i> XOR A,@(M+Ri)	XOR <i>A,(MR,SP)</i> XOR A,@(M+SP)	XOR <i>A,[MR.w]</i> XOR A,@M	XOR <i>A,[MN.w]</i> XOR A,@M	XOR <i>A,((MR.w),Ri)</i> XOR A,@(M+Ri)	XOR <i>A,((MN.w),X)</i> XOR A,@(M+X)
CP <i>A,#d8</i> CP A,#d8	CP <i>A,MR</i> CP A,M	CP <i>A,MN</i> CP A,M	CP <i>A,(Ri)</i> CP A,@Ri	CP <i>A,(MR,Ri)</i> CP A,@(M+Ri)	CP <i>A,(MN,Ri)</i> CP A,@(M+Ri)	CP <i>A,(MR,SP)</i> CP A,@(M+SP)	CP <i>A,[MR.w]</i> CP A,@M	CP <i>A,[MN.w]</i> CP A,@M	CP <i>A,((MR.w),Ri)</i> CP A,@(M+Ri)	CP <i>A,((MN.w),X)</i> CP A,@(M+X)
CPW <i>Ri,#d16</i> CPW Ri,#d16	CPW <i>Ri,MR</i> CPW Ri,M	CPW <i>Ri,MN</i> CPW Ri,M	CPW <i>Ri_a,(Ri_b)</i> CPW Ri_a,@Ri_b	CPW <i>Ri_a,(MR,Ri_b)</i> CPW Ri_a,@(M+Ri_b)	CPW <i>Ri_a,(MN,Ri_b)</i> CPW Ri_a,@(M+Ri_b)	CPW <i>X,(MR,SP)</i> CPW X,@(M+SP)	CPW <i>Ri,[MR.w]</i> CPW Ri,@M	CPW <i>X,[MN.w]</i> CPW X,@M		
BCP <i>A,#d8</i> BCP A,#d8	BCP <i>A,MR</i> BCP A,M	BCP <i>A,MN</i> BCP A,M	BCP <i>A,(Ri)</i> BCP A,@Ri	BCP <i>A,(MR,Ri)</i> BCP A,@(M+Ri)	BCP <i>A,(MN,Ri)</i> BCP A,@(M+Ri)	BCP <i>A,(MR,SP)</i> BCP A,@(M+SP)	BCP <i>A,[MR.w]</i> BCP A,@M	BCP <i>A,[MN.w]</i> BCP A,@M	BCP <i>A,((MR.w),Ri)</i> BCP A,@(M+Ri)	BCP <i>A,((MN.w),X)</i> BCP A,@(M+X)

ADC <i>A,#d8</i> ADC A,#d8	ADC <i>A,MR</i> ADC A,M	ADC <i>A,MN</i> ADC A,M	ADC <i>A,(Ri)</i> ADC A,@Ri	ADC <i>A,(MR,Ri)</i> ADC A,@(M+Ri)	ADC <i>A,(MN,Ri)</i> ADC A,@(M+Ri)	ADC <i>A,(MR,SP)</i> ADC A,@(M+SP)	ADC <i>A,[MR.w]</i> ADC A,@M	ADC <i>A,[MN.w]</i> ADC A,@M	ADC <i>A,((MR.w),Ri)</i> ADC A,@(M+Ri)	ADC <i>A,((MN.w),X)</i> ADC A,@(M+X)
ADD <i>A,#d8</i> ADD A,#d8	ADD <i>A,MR</i> ADD A,M	ADD <i>A,MN</i> ADD A,M	ADD <i>A,(Ri)</i> ADD A,@Ri	ADD <i>A,(MR,Ri)</i> ADD A,@(M+Ri)	ADD <i>A,(MN,Ri)</i> ADD A,@(M+Ri)	ADD <i>A,(MR,SP)</i> ADD A,@(M+SP)	ADD <i>A,[MR.w]</i> ADD A,@M	ADD <i>A,[MN.w]</i> ADD A,@M	ADD <i>A,((MR.w),Ri)</i> ADD A,@(M+Ri)	ADD <i>A,((MN.w),X)</i> ADD A,@(M+X)
SBC <i>A,#d8</i> SBC A,#d8	SBC <i>A,MR</i> SBC A,M	SBC <i>A,MN</i> SBC A,M	SBC <i>A,(Ri)</i> SBC A,@Ri	SBC <i>A,(MR,Ri)</i> SBC A,@(M+Ri)	SBC <i>A,(MN,Ri)</i> SBC A,@(M+Ri)	SBC <i>A,(MR,SP)</i> SBC A,@(M+SP)	SBC <i>A,[MR.w]</i> SBC A,@M	SBC <i>A,[MN.w]</i> SBC A,@M	SBC <i>A,((MR.w),Ri)</i> SBC A,@(M+Ri)	SBC <i>A,((MN.w),X)</i> SBC A,@(M+X)
SUB <i>A,#d8</i> SUB A,#d8	SUB <i>A,MR</i> SUB A,M	SUB <i>A,MN</i> SUB A,M	SUB <i>A,(Ri)</i> SUB A,@Ri	SUB <i>A,(MR,Ri)</i> SUB A,@(M+Ri)	SUB <i>A,(MN,Ri)</i> SUB A,@(M+Ri)	SUB <i>A,(MR,SP)</i> SUB A,@(M+SP)	SUB <i>A,[MR.w]</i> SUB A,@M	SUB <i>A,[MN.w]</i> SUB A,@M	SUB <i>A,((MR.w),Ri)</i> SUB A,@(M+Ri)	SUB <i>A,((MN.w),X)</i> SUB A,@(M+X)
ADDW <i>SP,#d8</i> ADDW SP,#d8	ADDW <i>Ri,#d16</i> ADDW Ri,#d16	ADDW <i>Ri,MR</i> ADDW Ri,M	ADDW <i>Ri,MN</i> ADDW Ri,M	ADDW <i>Ri,(MR,SP)</i> ADDW Ri,@(M+SP)		SUBW <i>SP,#d8</i> SUBW SP,#d8	SUBW <i>Ri,#d16</i> SUBW Ri,#d16	SUBW <i>Ri,MR</i> SUBW Ri,M	SUBW <i>Ri,MN</i> SUBW Ri,M	SUBW <i>Ri,(MR,SP)</i> SUBW Ri,@(M+SP)

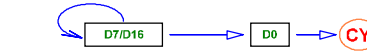
JRC <i>rel</i> JC rel C=1	JREQ <i>rel</i> JZ rel Z=1	JRH <i>rel</i> JRH rel H=1	JRV <i>rel</i> JRV rel V=1	JRPL <i>rel</i> J plus,rel >=0 N=0	JRM <i>rel</i> J interrupt mask rel !=1
JRNC <i>rel</i> JNC rel C=0	JRNE <i>rel</i> J !=,rel <>0 Z=0	JRNH <i>rel</i> JRNH rel H=0	JRNV <i>rel</i> JRV rel V=0	JRMI <i>rel</i> J minus,rel <0 N=1	JRNM <i>rel</i> J Not interrupt mask rel !=0

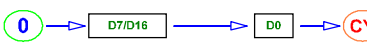
JRIH <i>rel</i> J interrupt line is high rel
JRIL <i>rel</i> J interrupt line is low rel

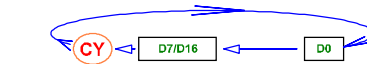
JRT <i>rel</i> J true, rel
JRF <i>rel</i> J false, rel

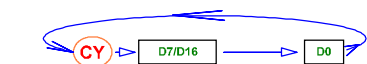
JRSGE <i>rel</i> rel signed greater or equal (N xor V)=0	JRSGT <i>rel</i> rel signed greater than (N xor V)=0	JRSLE <i>rel</i> rel signed lower or equal (N xor V)=1	JRSLT <i>rel</i> rel signed lower than (N xor V)=1
JRUGE <i>rel</i> rel unsigned greater or equal C=0	JRUGT <i>rel</i> rel unsigned greater than C=0 and Z=0	JRULE <i>rel</i> rel unsigned lower or equal C=1 or Z=1	JRULT <i>rel</i> rel unsigned lower than C=1

 **SLL/SLA ,SLLW/SLAW**

 **SRA ,SRAW**

 **SRL ,SRLW**

 **RLC ,RLCW**

 **RRC ,RRCW**

RLWA <i>Ri,A</i> A<~RiH<~RiL<~A	RRWA <i>Ri,A</i> A>~RiH>~RiL>~A
--	--

#bit - номер бита 0-7
#d8 - непосредственные данные 0-0xFF
#d16 - непосредственные данные 0-0xFFFF

rel -короткий переход +127/-128 относительно текущего значения PCE:PCH:PCL

MR диапазон адресов 0x000000-0x0000FF

M0 диапазон адресов 0x000000-0x00FFFF частный случай от MN

MN диапазон адресов 0xnn0000-0xnnFFFF

MF диапазон адресов 0x000000-0xFFFFFFF

В качестве MR,M0,MN,MF может использоваться прямоугонный адрес или метки определенной переоб 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)

(.....) - адрес ячейки памяти
[.....] - содержимое указателя по адресу M...

RM - 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 (е основном для директивы DC. или DS.)

BCP=AND, но **содержимое A НЕИЗМЕНЯЕТ**
выставляет флаги N и Z по результату