

AT mega memory map

© EVORAN Budapest Tel.: (361) 3-128-802

This table can be copied, distributed, used in any publication without any preliminary permission as long as this notice is not deleted or modified.

Port	Port	Addr	SRAM AREA	60h...0FFFh
Reg	Port	Addr	R64...R95 = PORT31...PORT63	40h...5Fh
	Port	Addr	R32...R63 = PORT0...PORT31	20h...3Fh
	Reg	Addr	R16...R31	10h...1Fh
	Reg	Addr	R0...R16	0...0fh

Rd=register destination Pd=port destination bit=0...7 range k=0...63 range
 Rs=register source Ps=port source byte=0...255 range offset= -64...+63 range
 Rp=register pair

note indicates if a statement acts on a particular register

Statement	Description	Cycles	Flags
ADC	Rd,Rs	1	Z,C,N,V,H
ADD	Rd,Rs	1	Z,C,N,V,H
ADIW	Rp,k Rp=N,X,Y,Z N=25:24 X=27:26 Y=29:28 Z=31:30 k=0...63	2	Z,C,N,V,S
AND	Rd,Rs	1	Z,N,V
ANDI	Rd,byte	1	Z,N,V
ASR	Rd Arithmetic Shift Right b7 > b7 b6 b5 b4 b3 b2 b1 b0 > C	1	Z,C,N,V
BCLR	bit Bit CleaR (in SREG)	1	defined bit
BLD	Rd, bit Bit LoaD (SREG:T > Rd:bit)	1	-
BRBC	bit,offset BRanch if Bit Clear (in SREG) offset= -64...+63	1/2	-
BRBS	bit,offset BRanch if Bit Set (in SREG) offset= -64...+63	1/2	-
BRCC	offset BRanch if Carry Cleared offset= -64...+63	1/2	-
BRCS	offset BRanch if Carry Set offset= -64...+63	1/2	-
BREQ	offset BRanch if EQual offset= -64...+63	1/2	-
BRGE	offset BRanch if Geather or Equal offset= -64...+63	1/2	-
BRHC	offset BRanch if Halfcarry Cleared offset= -64...+63	1/2	-
BRHS	offset BRanch if Halfcarry Set offset= -64...+63	1/2	-
BRID	offset BRanch if Interrupt Disabled offset= -64...+63	1/2	-
BRIE	offset BRanch if Interrupt Enabled offset= -64...+63	1/2	-
BRLO	offset BRanch if Lower offset= -64...+63	1/2	-
BRLT	offset BRanch if Less Than offset= -64...+63	1/2	-
BRMI	offset BRanch if Minus offset= -64...+63	1/2	-
BRNE	offset BRanch if Not Equal offset= -64...+63	1/2	-
BREQ	offset BRanch if EQual offset= -64...+63	1/2	-
BRPL	offset BRanch if PLus offset= -64...+63	1/2	-
BRSH	offset BRanch if Same or Higher offset= -64...+63	1/2	-
BRTC	offset BRanch if T flag Cleared offset= -64...+63	1/2	-
BRTS	offset BRanch if T flag Set offset= -64...+63	1/2	-
BRTC	offset BRanch if T flag Cleared offset= -64...+63	1/2	-
BRVC	offset BRanch if oVerflow Cleared offset= -64...+63	1/2	-
BRVS	offset BRanch if oVerflow Set offset= -64...+63	1/2	-
BSET	bit Bit SET (in SREG)	1	defined bit
BST	Rd, bit Bit STore (Rd:bit > SREG:T)	1	T
CALL	Addr Addr =0...65535	4	-
CBI	Pd, bit Clear Bit in I/o	2	-
CBR	Rd, byte Clear Bit in Register clears all bit positions, where byte is set	1	Z,N,V
CLC	CLear Carry	1	C
CLH	CLear Halfcarry	1	H
CLI	CLear Interrupt	1	I
CLN	CLear Negative flag	1	N
CLR	Rd CLear Register	1	Z,N,V
CLS	CLear Signed flag	1	S
CLT	CLear T flag	1	T
CLV	CLear oVerflow flag	1	V
CLZ	CLear Zero flag	1	Z
COM	Rd COMplement (one's complement) Rd=0FFh-Rd	1	Z,C,N,V
CP	Rd,Rs ComPare	1	Z,C,N,V,H
CPC	Rd,Rs ComPare with Carry Rd - Rs - C	1	Z,C,N,V,H
CPI	Rd,byte ComPare with Immediate	1	Z,C,N,V,H
CPSE	Rd,Rs ComPare, Skip if Equal	1/2	-
DEC	Rd	1	Z,N,V
EOR	Rd,Rs Exclusive OR	1	Z,N,V
ICALL	Indirect CALL call to address pointed by Z (R31:R30)	3	-
JMP	Indirect JuMP jump to address pointed by Z (R31:R30)	2	-
IN	Rd,Ps	1	-
INC	Rd	1	Z,N,V
JMP	Addr Addr =0...65535	3	-
LD	Rd,X LoaD into register from SRAM area pointed by X (R27:R26)	2	-
LD	Rd,X+ LoaD into register from SRAM area pointed by X (R27:R26), then incr. X	2	-
LD	Rd,-X decr. X, then LoaD into register from SRAM area pointed by X (R27:R26)	2	-
LD	Rd,Y LoaD into register from SRAM area pointed by Y (R29:R28)	2	-

LD	Rd,Y+	LoaD into register from SRAM area pointed by Y (R29:R28), then incr. Y	2	-
LD	Rd,-Y	decr. Y, then LoaD into register from SRAM area pointed by Y (R29:R28)	2	-
LDD	Rd,Y+k	LoaD with Displacement pointed by Y+k (R29:R28) k=0..63	2	-
LD	Rd,Z	LoaD into register from SRAM area pointed by Z (R31:R30)	2	-
LD	Rd,Z+	LoaD into register from SRAM area pointed by Z (R31:R30), then incr. Z	2	-
LD	Rd,-Z	decr. Z, then LoaD into register from SRAM area pointed by Z (R31:R30)	2	-
LDD	Rd,Z+k	LoaD with Displacement pointed by Z+k (R31:R30) k=0..63	2	-
LDI	Rd,byte	LoaD Immediate	1	-
LDS	Rd,Addr	LoaD Directly from SRAM (from location pointed by Addr)	3	-
LPM		Load from Program Memory (Z) > R0 (Z=R31:R30)	3	-
LSL	Rd	Logical Shift Left C < b7 b6 b5 b4 b3 b2 b1 b0 < 0	1	Z,C,N,V
LSR	Rd	Logical Shift Right 0 > b7 b6 b5 b4 b3 b2 b1 b0 > C	1	Z,C,N,V
MOV	Rd,Rs		1	-
MUL	Rd,Rs	MULTiPLY result in R1:R0 pair	2	C
NEG	Rd	NEGate (two's complement) Rd=0 - Rd	1	Z,C,N,V,H
NOP		No OPeration	1	-
OR	Rd,Rs	inclusive OR	1	Z,N,V
ORI	Rd,byte	inclusive OR with Immediate	1	Z,N,V
OUT	Pd,Rs		1	-
POP	Rd	POPp a single byte from stack	2	-
PUSH	Rd	PUSHes a single byte into the stack	2	-
RCALL	Addr	Relative CALL Addr =+/- 2kB faster then CALL and uses less space	3	-
RET		RETurn from subroutine	4	-
RETI		RETurn from Interrupt	4	I
RJMP	Addr	Relative JUMP Addr =+/- 2kB faster then JMP and uses less space	2	-
ROL	Rd	ROtate Left C < b7 b6 b5 b4 b3 b2 b1 b0 < C 1	1	Z,C,N,V
ROR	Rd	ROtate Right C > b7 b6 b5 b4 b3 b2 b1 b0 > C	1	Z,C,N,V
SBC	Rd,Rs	SuBtract with Carry	1	Z,C,N,V,H
SBCI	Rd,byte	SuBstract with Carry Immediate	1	Z,C,N,V,H
SBI	Pd, bit	Set Bit in I/o	2	-
SBIC	Pd, bit	Skip if Bit in I/o Cleared	1/2	-
SBIS	Pd, bit	Skip if Bit in I/o Set	1/2	-
SBIW	Rp,k	Rp=N,X,Y,Z N=25:24 X=27:26 Y=29:28 Z=31:30 k=0..63	2	Z,C,N,V,S
SBR	Rd, byte	Set Bit in Register sets all bit positions, where byte is set	1	Z,N,V
SBRC	Rd, bit	Skip if Bit in Register Cleared	1/2	-
SBRS	Rd, bit	Skip if Bit in Register Set	1/2	-
SEC		SEt Carry	1	C
SEH		SEt Halfcarry	1	H
SEI		SEt Interrupt	1	I
SEN		SEt Negative flag	1	N
SER	Rd	SEt Register bits	1	Z,N,V
SES		SEt Signed flag	1	S
SET		SEt T flag	1	T
SEV		SEt oVerflow flag	1	V
SEZ		SEt Zero flag	1	Z
SLEEP			1	-
ST	X,Rs	STore into SRAM area at address pointed by X (R27:R26)	2	-
ST	X+,Rs	STore into SRAM area at address pointed by X (R27:R26), then incr. X	2	-
ST	-X,Rs	decr. X, then STore into SRAM area pointed by X (R27:R26)	2	-
ST	Y,Rs	STore into SRAM area at address pointed by Y (R29:R28)	2	-
ST	Y+,Rs	STore into SRAM area at address pointed by Y (R29:R28), then incr. Y	2	-
ST	-Y,Rs	decr. Y, then STore into SRAM area at address pointed by Y (R29:R28)	2	-
STD	Y+k,Rs	STore with Displacement pointed by Y+k (R29:R28) k=0..63	2	-
ST	Z,Rs	STore into SRAM area at address pointed by Z (R31:R30)	2	-
ST	Z+,Rs	STore into SRAM area at address pointed by Z (R31:R30), then incr. Z2	-	-
ST	-Z,Rs	decr. Z, then STore into SRAM area at address pointed by Z (R31:R30)	2	-
STD	Z+k,Rs	STore with Displacement pointed by Z+k (R31:R30) k=0..63	2	-
STS	Addr,Rs	STore into SRAM (at location pointed by Addr)	3	-
SUB	Rd,Rs	SUBstract	1	Z,C,N,V,H
SUBI	Rd,byte	SUBstract Immediate	1	Z,C,N,V,H
SWAP	Rs	SWAPs 4 bit nibbles: bit7..bit4 with bit3..bit0	1	-
TST	Rs	TeSTs register (plus, minus or zero)	1	Z,N,V
WDR		WatchDog Reset	1	-