

Table 1. AT89 Instruction Set Summary⁽¹⁾

Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS			
ADD	A,R _n	1	12
ADD	A,direct	2	12
ADD	A,@R _i	1	12
ADD	A,#data	2	12
ADDC	A,R _n	1	12
ADDC	A,direct	2	12
ADDC	A,@R _i	1	12
ADDC	A,#data	2	12
SUBB	A,R _n	1	12
SUBB	A,direct	2	12
SUBB	A,@R _i	1	12
SUBB	A,#data	2	12
INC	A	1	12
INC	R _n	1	12
INC	direct	2	12
INC	@R _i	1	12
DEC	A	1	12
DEC	R _n	1	12
DEC	direct	2	12
DEC	@R _i	1	12
INC	DPTR	1	24
MUL	AB	1	48
DIV	AB	1	48
DA	A	1	12

Note: 1. All mnemonics copyrighted © Intel Corp., 1980.

Mnemonic	Description	Byte	Oscillator Period
LOGICAL OPERATIONS			
ANL	A,R _n	1	12
ANL	A,direct	2	12
ANL	A,@R _i	1	12
ANL	A,#data	2	12
ANL	direct,A	2	12
ANL	direct,#data	3	24
ORL	A,R _n	1	12
ORL	A,direct	2	12
ORL	A,@R _i	1	12
ORL	A,#data	2	12
ORL	direct,A	2	12
ORL	direct,#data	3	24
XRL	A,R _n	1	12
XRL	A,direct	2	12
XRL	A,@R _i	1	12
XRL	A,#data	2	12
XRL	direct,A	2	12
XRL	direct,#data	3	24
CLR	A	1	12
CPL	A	1	12
RL	A	1	12
RLC	A	1	12
LOGICAL OPERATIONS (continued)			

Mnemonic		Description	Byte	Oscillator Period
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,R _n	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@R _i	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	R _n ,A	Move Accumulator to register	1	12
MOV	R _n ,direct	Move direct byte to register	2	24
MOV	R _n ,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,R _n	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@R _i	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@R _i ,A	Move Accumulator to indirect RAM	1	12
MOV	@R _i ,direct	Move direct byte to indirect RAM	2	24
MOV	@R _i ,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@R _i	Move External RAM (8-bit addr) to Acc	1	24
DATA TRANSFER (continued)				

Mnemonic		Description	Byte	Oscillator Period
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@R _i ,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,R _n	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@R _i	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@R _i	Exchange low-order Digit indirect RAM with Acc	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				

Mnemonic		Description	Byte	Oscillator Period
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	R _n ,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@R _i ,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	R _n ,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12