

8051 Instruction set

Abbreviations:

- direct = 8-bit DATA address in internal memory
- data = 8-bit constant in CODE memory
- data16 = 16-bit constant in CODE memory
- addr16 = 16-bit long CODE address
- addr11 = 11-bit absolute CODE address
- rel = signed 8-bit relative CODE address
- bit = 8-bit BIT address in internal memory
- Ri = register numbers 0 or 1

Rn = register numbers 0 thru 7

DATA TRANSFER

Instruction	Description	Bytes	Periods	C	OV	AC
MOV A,Rn	Move Register to ACC	1	12			
MOV A,direct	Move Direct byte to ACC	2	12			
MOV A,@Ri	Move Indirect byte to ACC	1	12			
MOV A,#data	Move Immediate data to ACC	2	12			
MOV Rn,A	Mov ACC to Register	1	12			
MOV Rn,direct	Move Direct byte to Register	2	24			
MOV Rn,#data	Move Immediate data to Register	2	12			
MOV direct,A	Move ACC to Direct byte	2	12			
MOV direct,Rn	Move Register to Direct byte	2	24			
MOV direct,direct	Move Direct byte to Direct byte	3	24			
MOV direct,@Ri	Mov Indirect RAM to Direct byte	3	24			
MOV direct,#data	Move Immediate data to Direct byte	3	24			
MOV @Ri,A	Move ACC to Indirect RAM	1	12			
MOV @Ri,direct	Move direct byte to indirect RAM.	2	24			
MOV @Ri,#data	Move Immediate data to Indirect RAM	2	12			
MOV DPTR,#data16	Load datapointer with 16 bit constant	3	24			
MOVC A,@A+DPTR	Move code byte at ACC+DPTR to ACC	1	24			
MOVC A,@A+PC	Move code byte at ACC+PC to ACC	1	24			
MOVX A,@Ri	Move external RAM to ACC	1	24			
MOVX @Ri,A	Move ACC to external RAM	1	24			
MOVX A,@DPTR	Move external RAM to ACC	1	24			
MOVX @DPTR,A	Move ACC to external RAM	1	24			
PUSH direct	Push direct byte to stack	2	24			
POP direct	Pop direct byte from stack	2	24			
XCH A,Rn	Exchange register with ACC	1	12			
XCH A,direct	Exchange direct byte with ACC	2	12			
XCH A,@Ri	Exchange indirect RAM with ACC	1	12			
XCHD A,@Ri	Exchange low order digit indirect RAM with ACC	1	12			

ARITHMETIC OPERATORS

Instruction	Description	Bytes	Periods	C	OV	AC
ADD A, Rn	Add register to ACC	1	12	x	x	x
ADD A, direct	Add direct byte to ACC	2	12	x	x	x
ADD A, @Ri	Add indirect RAM to ACC	1	12	x	x	x
ADD A, #data	Add immediate data to ACC	2	12	x	x	x
ADDC A, Rn	Add register to ACC with Carry	1	12	x	x	x
ADDC A, direct	Add direct byte to ACC with Carry	2	12	x	x	x
ADDC A, @Ri	Add indirect RAM to ACC with Carry	1	12	x	x	x
ADDC A, #data	Add immediate data to ACC with Carry	2	12	x	x	x
SUBB A, Rn	Subtract Register from ACC with borrow	1	12	x	x	x
SUBB A, direct	Subtract indirect RAM from ACC with borrow	2	12	x	x	x
SUBB A, @Ri	Subtract indirect RAM from ACC with borrow	1	12	x	x	x
SUBB A, #data	Subtract immediate data from ACC with borrow	2	12	x	x	x
INC A	Increment ACC	1	12			
INC Rn	Increment register	1	12			
INC direct	Increment direct byte	2	12			
INC @Ri	Increment direct RAM	1	12			
DEC A	Decrement ACC	1	12			
DEC Rn	Decrement Register	1	12			
DEC direct	Decrement direct byte	2	12			
DEC @Ri	Decrement indirect RAM	1	12			
INC DPTR	Increment Data Pointer	1	24			
MUL AB	Multiply A & B	1	48	0	x	
DIV AB	Divide A by B	1	48	0	x	
DA A	Decimal Adjust ACC	1	12	x		

LOGICAL OPERATIONS

Instruction	Description	Bytes	Periods	C	OV	AC
ANL A,Rn	AND register to ACC	1	12			
ANL A,direct	AND direct byte to ACC	2	12			
ANL A,@Ri	AND indirect RAM to ACC	1	12			
ANL A,#data	AND immediate data to ACC	2	12			
ANL direct,A	AND ACC to direct byte	2	12			
ANL direct,#data	AND immediate data to direct byte	3	24			
ORL A,Rn	OR register to ACC	1	12			
ORL A,direct	OR direct byte to ACC	2	12			
ORL A,@Ri	OR indirect RAM to ACC	1	12			
ORL A,#data	OR immediate data to ACC	2	12			
ORL direct,A	OR ACC to direct byte	2	12			
ORL direct,#data	OR immediate data to direct byte	3	24			
XRL A,Rn	XOR register to ACC	1	12			
XRL A,direct	XOR direct byte to ACC	2	12			
XRL A,@Ri	XOR indirect RAM to ACC	1	12			
XRL A,#data	XOR immediate data to ACC	2	12			
XRL direct,A	XOR ACC to direct byte	2	12			
XRL direct,#data	XOR immediate data to direct byte	3	24			
CLR A	Clear the ACC	1	12			
CPL A	Complement the ACC	1	12			
RL A	Rotate the ACC left	1	12			
RLC A	Rotate the ACC left through Carry	1	12	x		
RR A	Rotate the ACC right	1	12			
RRC A	Rotate the ACC right through Carry	1	12	x		
SWAP A	Swap nibbles in the ACC	1	12			

BOOLEAN MANIPULATION

Instruction	Description	Bytes	Periods	C	OV	AC
CLR C	Clear carry flag	1	12	0		
CLR bit	Clear direct bit	2	12			
SETB C	Set carry flag	1	12	1		
SETB bit	Set direct bit	2	12			
CPL C	Complement carry flag	1	12	x		
CPL bit	Complement direct bit	2	12			
ANL C,bit	AND direct bit to carry	2	24	x		
ANL C,/bit	AND complement of direct bit to carry	2	24	x		
ORL C,bit	OR direct bit to carry	2	24	x		
ORL C,/bit	OR complement of direct bit to carry	2	24	x		
MOV C,bit	Move direct bit to carry	2	12	x		
MOV bit,C	Move carry to direct bit	2	24			
JC rel	Jump if carry is set	2	24			
JNC rel	Jump if carry is 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 and clear that bit	3	24			

PROGRAM BRANCHING

Instruction	Description	Bytes	Periods	C	OV	AC
ACALL addr11	Absolute call within 2K page	2	24			
LCALL addr16	Absolute call (Long call)	3	24			
RET	Return from subroutine	1	24			
RETI	Return from interrupt	1	24			
AJMP addr11	Absolute jump within 2K page	2	24			
LJMP addr16	Absolute jump (Long jump)	3	24			
SJMP rel8	Relative jump within +/- 127 bytes (Short jump)	2	24			
JMP @A+DPTR	Jump direct relative to DPTR	1	24			
JZ rel8	Jump if ACC is zero	2	24			
JNZ rel8	Jump if ACC is NOT zero	2	24			
CJNE A,direct,rel8	Compare direct byte to ACC, jump if NOT equal	3	24	x		
CJNE A,#data,rel8	Compare immediate to ACC, jump if NOT equal	3	24	x		
CJNE Rn,#data,rel8	Compare immediate to register, jump if NOT equal	3	24	x		
CJNE @Ri,#data,rel8	Compare immediate to indirect, jump if NOT equal	3	24	x		
DJNZ Rn,rel8	Decrement register, jump if NOT zero	2	24			
DJNZ direct,rel8	Decrement direct byte, jump if NOT zero	3	24			
NOP	No operation (Skip to next instruction)	1	12			