DATA TRANSFER	*ARITHMETIC*	*LOGICAL*	*BOOLEAN*	*PROGRAM BRANCHING*
MOV	ADD	ANL	CLR	LJMP
MOVC	ADDC	ORL	SETB	AJMP
MOVX	SUBB	XRL	MOV	SJMP
PUSH	INC	CLR	JC	JZ
POP	DEC	CPL	JNC	JNZ
XCH	MUL	RL	JB	CJNE
XCHD	DIV	RLC	JNB	DJNZ
	DA A	RR	JBC	NOP
		RRC	ANL	LCALL
		SWAP	ORL	ACALL
			CPL	RET
				RETI
				JMP

Instruction Set of 8051 Microcontroller

instruction set of 8051 microcontrollers. Complete information regarding each instruction like operational explanation, addressing mode, no. of byte occupied, no. of cycles used etc is given.

Data moving / handling Instructions:

Mnemonics	Operational description	Addressing mode	No. of bytes occupied	No. of cycles used
Mov a,#num	Copy the immediate data num in to acc	immediate	2	1
Mov Rx,a	Copy the data from acc to Rx	register	1	1
Mov a,Rx	Copy the data from Rx to acc	register	1	1
Mov Rx,#num	Copy the immediate data num in to Rx	immediate	2	1

Mov a,add	Copy the data from direct address add to acc	direct	2	1
Mov add,a	Copy the data from acc to direct address add	direct	2	1
Mov add,#num	Copy the immediate data num in to direct address	direct	3	2
Mov Rx,add	Copy the data from direct address add to Rx	direct	2	2
Mov add,Rx	Copy the data from Rx to direct address add	direct	2	2
Mov @Rp,a	Copy the data in acc to address in Rp	Indirect	1	1
Mov a,@Rp	Copy the data that is at address in Rp to acc	Indirect	1	1
Mov add,@Rp	Copy the data that is at address in Rp to add	Indirect	2	2
Mov @Rp,add	Copy the data in add to address in Rp	Indirect	2	2
Mov @Rp,#num	Copy the immediate byte num to the address in Rp	Indirect	2	1
Movx a,@Rp	Copy the content of external add in Rp to acc	Indirect	1	2
Movx a,@DPTR	Copy the content of external add in DPTR to acc	Indirect	1	2
Movx @Rp,a	Copy the content of acc to the external add in Rp	Indirect	1	2

Movx @DPTR,a	Copy the content of acc to the external add in DPTR	Indirect	1	2
Movc a,@a+DPTR	The **MOVC** instruction moves a byte from the code or program memory to the accumulator. The address is formed by adding acc and DPTR and its content is copied to acc	indirect	1	2
Movc a, @a+PC	The address is formed by adding acc and PC and its content is copied to acc	indirect	1	2
Push add	Increment SP and copy the data from source add to internal RAM address contained in SP	Direct	2	2
Pop add	copy the data from internal RAM address contained in SP to destination add and decrement SP	direct	2	2
Xch a, Rx	Exchange the data between acc and Rx	Register	1	1
Xch a, add	Exchange the data between acc and given add	Direct	2	1
Xch a,@Rp	Exchange the data between acc and address in Rp	Indirect	1	1

Xchd a, @Rp	Exchange only lower nibble of acc and address in Rp	indirect	1	1

Logical Instructions

Mnemonics	Operational description	Addressing mode	No. of bytes occupied	No. of cycles used
Anl a, #num	AND each bit of acc with same bit of immediate num, stores result in acc	Immediate	2	1
Anl a, add	AND each bit of acc with same bit of content in add, stores result in acc	Direct	2	1
Anl a, Rx	AND each bit of acc with same bit of content of Rx, stores result in acc	Register	1	1
Anl a, @Rp	AND each bit of acc with same bit of content of add given by Rp, stores result in acc	Indirect	1	1
Anl add, a	AND each bit of acc with same bit of direct add num, stores result in add	Direct	2	1
Anl add, #num	AND each bit of direct add with same bit of immediate num, stores result in add	direct	3	2

orl a, #num	OR each bit of acc with same bit of immediate num, stores result in acc	Immediate	2	1
orl a, add	OR each bit of acc with same bit of content in add, stores result in acc	Direct	2	1
orl a, Rx	OR each bit of acc with same bit of content of Rx, stores result in acc	Register	1	1
orl a, @Rp	OR each bit of acc with same bit of content of add given by Rp, stores result in acc	Indirect	1	1
orl add, a	OR each bit of acc with same bit of direct add num, stores result in add	Direct	2	1
orl add, #num	OR each bit of direct add with same bit of immediate num, stores result in add	direct	3	2
Xrl a, #num	XOR each bit of acc with same bit of immediate num, stores result in acc	Immediate	2	1
Xrl a, add	XOR each bit of acc with same bit of content in add, stores result in acc	Direct	2	1

Xrl a, Rx	XOR each bit of acc with same bit of content of Rx, stores result in acc	Register	1	1
Xrl a, @Rp	XOR each bit of acc with same bit of content of add given by Rp, stores result in acc	Indirect	1	1
Xrl add, a	XOR each bit of acc with same bit of direct add num, stores result in add	Direct	2	1
Xrl add, #num	XOR each bit of direct add with same bit of immediate num, stores result in add	direct	3	2
Clr a	Clear each bit of acc	Direct	1	1
Cpl a	Complement each bit of acc	direct	1	1
Anl c, b	AND carry with given bit b, stores result in carry	—	2	2
Anl c, /b	AND carry with complement of given bit b, stores result in carry	—	2	2
Orl c, b	OR carry with given bit b, stores result in carry	—	2	2
Orl c, /b	OR carry with complement of given bit b, stores result in carry	—	2	2
Cpl c	Complement carry flag	—	1	1

Cpl b	Complement bit b	—	2	1
Clr c	Clear carry flag	—	1	1
Clr b	Clear given bit b	—	2	1
Mov c, b	Copy bit b to carry	—	2	1
Mov b, c	Copy carry to bit b	—	2	2
Setb c	Set carry flag	—	1	1
Setb b	Set bit b	—	2	1
Rl a	Rotate acc one bit left	—	1	1
Rr a	Rotate acc one bit right	—	1	1
Rlc a	Rotate acc one bit left with carry	—	1	1
Rrc a	Rotate acc one bit right with carry	—	1	1
Swap a	Exchange upper and lower nibble of acc	—	1	1

Arithmetic Instructions

Mnemonics	Operational description	Addressing mode	No. of bytes occupied	No. of cycles used
Inc a	Add 1 to acc	Register	1	1
Inc Rr	Add 1 to register Rr	Register	1	1
Inc add	Add 1 to the content of add	Direct	2	1
Inc @rp	Add 1 to the content of the address in Rp	indirect	1	1
Inc DPTR	Add 1 to DPTR	Register	1	2
dec a	Subtract 1 from acc	Register	1	1
dec Rr	Subtract 1 from Rr	Register	1	1

dec add	Subtract 1 from content of add	Direct	2	1
dec @rp	Subtract 1 from the content of address	indirect	1	1
Add a, #num	Add the immediate num with acc and stores result in acc	immediate	2	1
Add a, Rx	Add the data in Rx with acc and stores result in acc	Register	1	1
Add a, add	Add the data in add with acc and stores result in acc	Direct	2	1
Add a, @Rp	Add the data at the address in Rp with acc and stores result in acc	Indirect	1	1
Addc a,#num	Add the immediate num with acc and carry, stores result in acc	immediate	2	1
Addc a, Rx	Add the data in Rx with acc and carry, stores result in acc	Register	1	1
Addc a, add	Add the data in add with acc and carry, stores result in acc	Direct	2	1
Addc a, @Rp	Add the data at the address in Rp with acc and carry, stores result in acc	Indirect	1	1
Subb a, #num	Subtract immediate num and carry from acc; stores the result in acc	immediate	2	1

Subb a, add	Subtract the content of add and carry from acc; stores the result in acc	Register	1	1
Subb a, Rx	Subtract the data in Rx and carry from acc; stores the result in acc	Direct	2	1
Subb a, @Rp	Subtract the data at the address in Rp and carry from acc; stores the result in acc	Indirect	1	1
Mul ab	Multiply acc and register B. store the lower byte of result in acc and higher byte in B	—	1	4
div ab	divide acc by register B. store quotient in acc and remainder in B	—	1	4
Da a	After addition of two packed BCD numbers, adjust the sum to decimal format	—	1	1

Branching Instructions

Mnemonic	Operational description	No of bytes occupied	No. of cycles used
Jc label	Jump to label if carry is set to 1	2	2
Jnc label	Jump to label if carry is cleared to 0	2	2
Jb b,label	Jump to label if given bit is set to 1	3	2

Jnb b,label	Jump to label if given bit is cleared to 0	3	2
Jbc b,label	Jump to label if given bit is set. Clear the bit	3	2
Cjne a, add, label	Compare the content of accumulator with the content of given address and if not equal jump to label	3	2
Cjne a, #num, label	Compare the content of accumulator with immediate number and if not equal jump to label	3	2
Cjne Rx, #num, label	Compare the content of Rx with the immediate number and if not equal jump to label	3	2
Cjne @Rp, #num, label	Compare the content of location in Rp with immediate number and if not equal jump to label	3	2
Djnz Rx, label	Decrement the content of Rx and jump to the label if it is not zero	2	2
Djnz add, label	Decrement the content of address and jump to the label if it is not zero	3	2
Jz label	Jump to the label if content of accumulator is 0	2	2
Jnz label	Jump to the label if content of accumulator is not 0	2	2
Jmp @a+dptr	Jump to the address created by adding the contents on accumulator and dptr	1	2

Ajmp sadd	Take a jump to absolute short range address sadd	2	2
Ljmp ladd	Take a jump to absolute long range address sadd	3	2
Sjmp radd	Take a jump to relative address radd	2	2
nop	Short form of no operation means do nothing and go to next instruction	1	1
Acall sadd	Pushes the content of Acc on stack and load it will absolute short range address sadd	2	2
Lcall ladd	Pushes the content of Acc on stack and load it will absolute long range address sadd	3	2
Ret	returns from subroutine by restoring the Acc from stack using pop operation	1	2
reti	Returns from interrupt subroutine by restoring Acc from stack using pop operation	1	2