



Instruction set

- Data Transfer Instructions
- Arithmetic Instructions
- Logical Instructions
- Branching Instructions
- Bit Manipulation Instructions



Data transfer instructions



Mnemonic	Operation	Description	Example
MOV	Move	Transfer data between registers/memory	MOV A, #55H
MOVC	Move Code Byte	Copy from Code memory to A	MOVC A, @A+DPTR
MOVX	Move External	Transfer between A and External RAM	MOVX A, @DPTR
PUSH	Push onto Stack	Save byte onto Stack	PUSH 30H
POP	Pop from Stack	Restore byte from Stack	POP 30H
XCH	Exchange	Swap data between A and memory/register	XCH A, R0
XCHD	Exchange Lower Nibble	Swap lower 4 bits of A and RAM[@Ri]	XCHD A, @R0
MOV A, Pn	Input from Port	Read data from Port 0–3 to A	MOV A, P1
MOV Pn, A	Output to Port	Write data from A to Port 0–3	MOV P2, A



Arithmetic Instructions



Mnemonic	Operation	Description	Example
ADD A, Rn	Add	Adds src to accumulator A	ADD A, #25H
ADDC A, Rn	Add with Carry	Adds src and carry flag to accumulator A	ADDC A, R2
SUBB A, Rn	Subtract with Borrow	Subtracts src + carry from A	SUBB A, #10H
INC dst	Increment	Increases value of dst by 1	INC R0
DEC dst	Decrement	Decreases value of dst by 1	DEC 40H
MUL AB	Multiply	Multiplies A $\times$ B; result in A (low), B (high)	MUL AB
DIV AB	Divide	Divides A $\div$ B; quotient in A, remainder in B	DIV AB
DA A	Decimal Adjust	Adjusts A after BCD addition	DA A (after ADD)



Logical Instructions

Instruction	Operation	Description	Example
ANL A, Rn	$A \leftarrow A \wedge Rn$	AND register with Accumulator	ANL A, R1A = A AND R1
ORL A, Rn	$A \leftarrow A \vee Rn$	OR register with Accumulator	ORL A, R2A = A OR R2
XRL A, Rn	$A \leftarrow A \oplus Rn$	XOR register with Accumulator	XRL A, R4A = A XOR R4
CLR A	$A \leftarrow 00H$	Clear Accumulator	CLR AA = 00H
CPL A	$A \leftarrow -A$	Complement Accumulator	CPL AA = NOT A
CLR C	$CY \leftarrow 0$	Clear Carry Flag	CLR CCY = 0
SETB C	$CY \leftarrow 1$	Set Carry Flag	SETB CCY = 1
CPL C	$CY \leftarrow -CY$	Complement Carry Flag	CPL CCY = NOT CY
CLR bit	$bit \leftarrow 0$	Clear bit address	CLR P2.1P2.1 = 0
SETB bit	$bit \leftarrow 1$	Set bit address	SETB P3.3P3.3 = 1
CPL bit	$bit \leftarrow -bit$	Complement bit address	CPL P1.0P1.0 = NOT P1.0



Branching Instructions



Instruction	Operation	Description	Example
SJMP rel	Short jump (8-bit offset)	Jump to a nearby location (± 127 bytes)	SJMP BACK Jump to label BACK
AJMP addr11	Absolute jump within 2K block	Jump to an address within the same 2K page	AJMP 0123H
LJMP addr16	Long jump (16-bit)	Jump anywhere in 64KB code memory	LJMP MAIN Jump to label MAIN
JC rel	Jump if Carry = 1	Conditional jump if CY is set	JC OVERFLOW
JNC rel	Jump if Carry = 0	Conditional jump if CY is clear	JNC SAFE
JZ rel	Jump if Accumulator = 0	Conditional jump if A == 0	JZ ZERO CASE
JNZ rel	Jump if Accumulator $\neq$ 0	Conditional jump if A $\neq$ 0	JNZ LOOP
JB bit, rel	Jump if bit = 1	Conditional jump if specified bit is set	JB P1.0, SKIP
JNB bit, rel	Jump if bit = 0	Conditional jump if specified bit is clear	JNB P3.5, WAIT
JBC bit, rel	Jump if bit = 1 and clear bit	Jump if bit = 1, then clears it	JBC 20H, NEXT



Branching Instructions



Instruction	Operation	Description	Example
CJNE A, #data, rel	Compare A with immediate, jump if not equal	CJNE A, #0FFH, NOTMATCH	
DJNZ Rn, rel	Decrement register and jump if not zero	Looping instruction	DJNZ R1, REPEAT
DJNZ direct, rel	Decrement direct byte and jump if not zero	DJNZ 20H, CONTINUE	
ACALL addr11	Absolute call within 2K block	Call subroutine in same 2K page	ACALL DELAY
LCALL addr16	Long call to subroutine	Call anywhere in 64KB memory	LCALL SUB_ROUTINE
RET	Return from subroutine	Returns to the calling function	RET
RETI	Return from interrupt	Used at end of ISR	RETI
NOP	No operation	Do nothing (1 cycle)	NOP



Bit Manipulation Instructions



Instruction	Operation	Description	Example
SETB bit	$\text{bit} \leftarrow 1$	Set the specified bit to 1	SETB P1.0 P1.0 = 1
CLR bit	$\text{bit} \leftarrow 0$	Clear the specified bit to 0	CLR P3.3 P3.3 = 0
CPL bit	$\text{bit} \leftarrow \neg \text{bit}$	Complement (toggle) the specified bit	CPL P2.5 P2.5 = NOT P2.5
ANL C, bit	$\text{CY} \leftarrow \text{CY} \wedge \text{bit}$	Logical AND bit with Carry	ANL C, P1.1
ORL C, bit	$\text{CY} \leftarrow \text{CY} \vee \text{bit}$	Logical OR bit with Carry	ORL C, ACC.7
MOV C, bit	$\text{CY} \leftarrow \text{bit}$	Move specified bit to Carry	MOV C, P1.0
MOV bit, C	$\text{bit} \leftarrow \text{CY}$	Move Carry to specified bit	MOV P2.1, C
CLR C	$\text{CY} \leftarrow 0$	Clear Carry flag	CLR C
SETB C	$\text{CY} \leftarrow 1$	Set Carry flag	SETB C
CPL C	$\text{CY} \leftarrow \neg \text{CY}$	Complement Carry flag	CPL C
JBC bit, rel	If bit = 1, clear it and jump	Used for flags and one-time events	JBC 30H, FLAG_DONE
JB bit, rel	If bit = 1, jump	Conditional bit test	JB P3.7, CHECKNEXT
JNB bit, rel	If bit = 0, jump	Conditional bit test	JNB ACC.0, LOOP