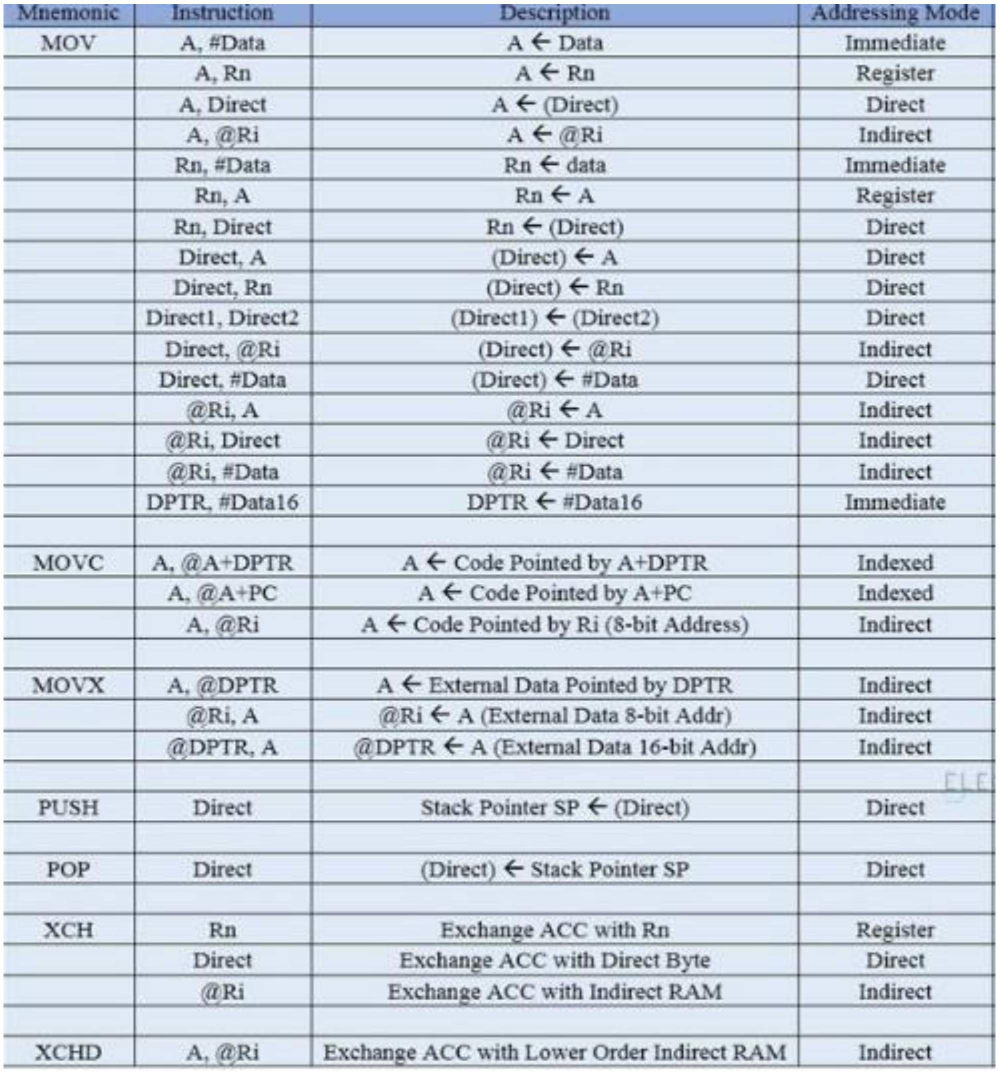
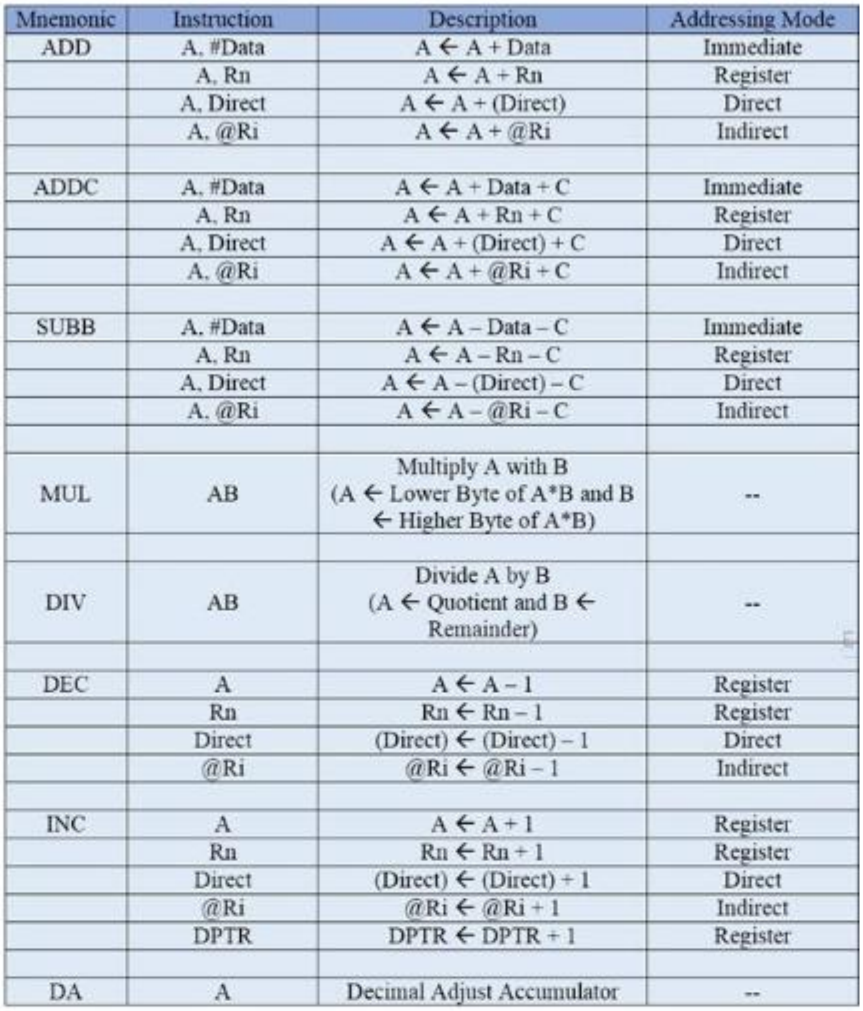


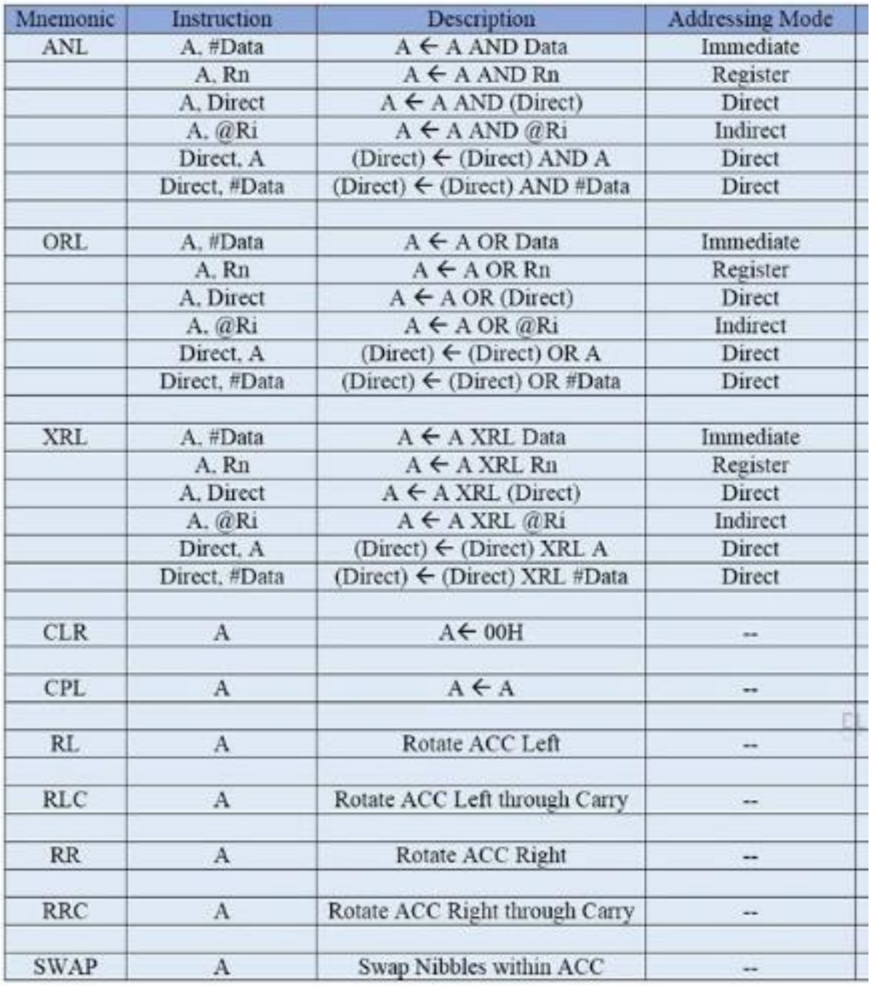
Mnemonic	Instruction	Description	Addressing Mode
MOV	A, #Data	A ← Data	Immediate
	A, Rn	A ← Rn	Register
	A, Direct	A ← (Direct)	Direct
	A, @Ri	A ← @Ri	Indirect
	Rn, #Data	Rn ← data	Immediate
	Rn, A	Rn ← A	Register
	Rn, Direct	Rn ← (Direct)	Direct
	Direct, A	(Direct) ← A	Direct
	Direct, Rn	(Direct) ← Rn	Direct
	Direct1, Direct2	(Direct1) ← (Direct2)	Direct
	Direct, @Ri	(Direct) ← @Ri	Indirect
	Direct, #Data	(Direct) ← #Data	Direct
	@Ri, A	@Ri ← A	Indirect
	@Ri, Direct	@Ri ← Direct	Indirect
	@Ri, #Data	@Ri ← #Data	Indirect
	DPTR, #Data16	DPTR ← #Data16	Immediate
MOVC	A, @A+DPTR	A ← Code Pointed by A+DPTR	Indexed
	A, @A+PC	A ← Code Pointed by A+PC	Indexed
	A, @Ri	A ← Code Pointed by Ri (8-bit Address)	Indirect
MOVX	A, @DPTR	A ← External Data Pointed by DPTR	Indirect
	@Ri, A	@Ri ← A (External Data 8-bit Addr)	Indirect
	@DPTR, A	@DPTR ← A (External Data 16-bit Addr)	Indirect
PUSH	Direct	Stack Pointer SP ← (Direct)	Direct
POP	Direct	(Direct) ← Stack Pointer SP	Direct
XCH	Rn	Exchange ACC with Rn	Register
	Direct	Exchange ACC with Direct Byte	Direct
	@Ri	Exchange ACC with Indirect RAM	Indirect
XCHD	A, @Ri	Exchange ACC with Lower Order Indirect RAM	Indirect

Data Transfer Instructions



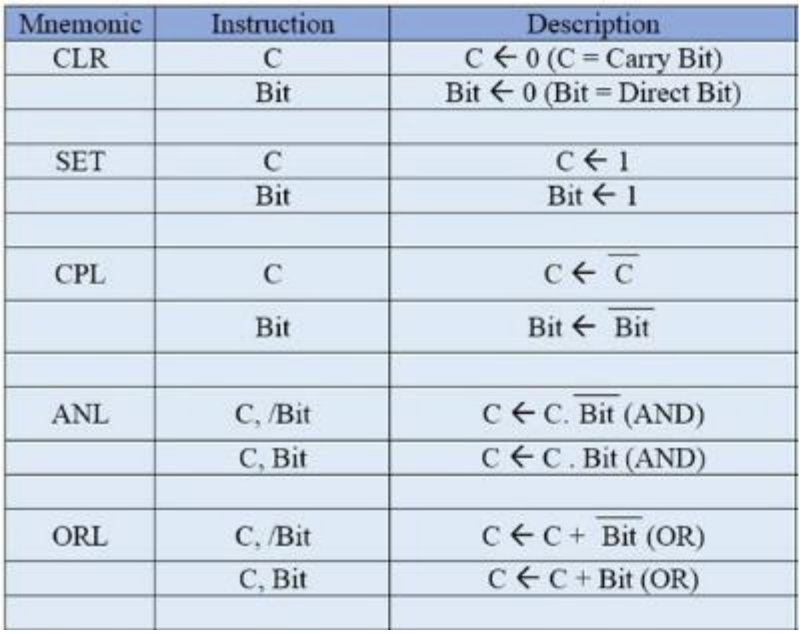
Mnemonic	Instruction	Description	Addressing Mode
ADD	A, #Data	A ← A + Data	Immediate
	A, Rn	A ← A + Rn	Register
	A, Direct	A ← A + (Direct)	Direct
	A, @Ri	A ← A + @Ri	Indirect
ADDC	A, #Data	A ← A + Data + C	Immediate
	A, Rn	A ← A + Rn + C	Register
	A, Direct	A ← A + (Direct) + C	Direct
	A, @Ri	A ← A + @Ri + C	Indirect
SUBB	A, #Data	A ← A – Data – C	Immediate
	A, Rn	A ← A – Rn – C	Register
	A, Direct	A ← A – (Direct) – C	Direct
	A, @Ri	A ← A – @Ri – C	Indirect
MUL	AB	Multiply A with B (A ← Lower Byte of A*B and B ← Higher Byte of A*B)	--
DIV	AB	Divide A by B (A ← Quotient and B ← Remainder)	--
DEC	A	A ← A – 1	Register
	Rn	Rn ← Rn – 1	Register
	Direct	(Direct) ← (Direct) – 1	Direct
	@Ri	@Ri ← @Ri – 1	Indirect
INC	A	A ← A + 1	Register
	Rn	Rn ← Rn + 1	Register
	Direct	(Direct) ← (Direct) + 1	Direct
	@Ri	@Ri ← @Ri + 1	Indirect
	DPTR	DPTR ← DPTR + 1	Register
DA	A	Decimal Adjust Accumulator	--

Arithmetic Instructions

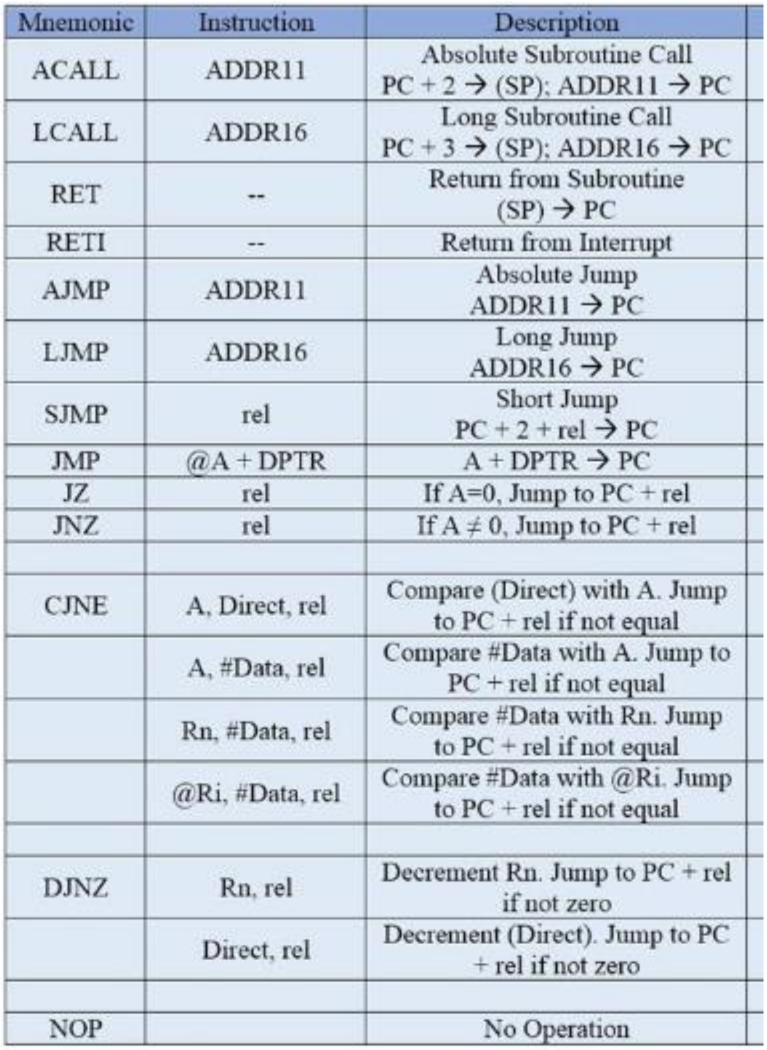


Mnemonic	Instruction	Description	Addressing Mode
ANL	A, #Data	A ← A AND Data	Immediate
	A, Rn	A ← A AND Rn	Register
	A, Direct	A ← A AND (Direct)	Direct
	A, @Ri	A ← A AND @Ri	Indirect
	Direct, A	(Direct) ← (Direct) AND A	Direct
	Direct, #Data	(Direct) ← (Direct) AND #Data	Direct
ORL	A, #Data	A ← A OR Data	Immediate
	A, Rn	A ← A OR Rn	Register
	A, Direct	A ← A OR (Direct)	Direct
	A, @Ri	A ← A OR @Ri	Indirect
	Direct, A	(Direct) ← (Direct) OR A	Direct
	Direct, #Data	(Direct) ← (Direct) OR #Data	Direct
XRL	A, #Data	A ← A XRL Data	Immediate
	A, Rn	A ← A XRL Rn	Register
	A, Direct	A ← A XRL (Direct)	Direct
	A, @Ri	A ← A XRL @Ri	Indirect
	Direct, A	(Direct) ← (Direct) XRL A	Direct
	Direct, #Data	(Direct) ← (Direct) XRL #Data	Direct
CLR	A	A← 00H	--
CPL	A	A ← A	--
RL	A	Rotate ACC Left	--
RLC	A	Rotate ACC Left through Carry	--
RR	A	Rotate ACC Right	--
RRC	A	Rotate ACC Right through Carry	--
SWAP	A	Swap Nibbles within ACC	--

Logical Instructions



Mnemonic	Instruction	Description
CLR	C	C ← 0 (C = Carry Bit)
	Bit	Bit ← 0 (Bit = Direct Bit)
SET	C	C ← 1
	Bit	Bit ← 1
CPL	C	C ← $\overline{\text{C}}$
	Bit	Bit ← $\overline{\text{Bit}}$
ANL	C, /Bit	C ← C. $\overline{\text{Bit}}$ (AND)
	C, Bit	C ← C . Bit (AND)
ORL	C, /Bit	C ← C + $\overline{\text{Bit}}$ (OR)
	C, Bit	C ← C + Bit (OR)



Mnemonic	Instruction	Description
ACALL	ADDR11	Absolute Subroutine Call PC + 2 → (SP); ADDR11 → PC
LCALL	ADDR16	Long Subroutine Call PC + 3 → (SP); ADDR16 → PC
RET	--	Return from Subroutine (SP) → PC
RETI	--	Return from Interrupt
AJMP	ADDR11	Absolute Jump ADDR11 → PC
LJMP	ADDR16	Long Jump ADDR16 → PC
SJMP	rel	Short Jump PC + 2 + rel → PC
JMP	@A + DPTR	A + DPTR → PC
JZ	rel	If A=0, Jump to PC + rel
JNZ	rel	If A ≠ 0, Jump to PC + rel
CJNE	A, Direct, rel	Compare (Direct) with A. Jump to PC + rel if not equal
	A, #Data, rel	Compare #Data with A. Jump to PC + rel if not equal
	Rn, #Data, rel	Compare #Data with Rn. Jump to PC + rel if not equal
	@Ri, #Data, rel	Compare #Data with @Ri. Jump to PC + rel if not equal
DJNZ	Rn, rel	Decrement Rn. Jump to PC + rel if not zero
	Direct, rel	Decrement (Direct). Jump to PC + rel if not zero
NOP		No Operation