

ADD	A,Meml	ADD	$A \leftarrow A + \text{Meml}$
ADC	A,Meml	ADD with Carry	$A \leftarrow A + \text{Meml} + C$, $C \leftarrow \text{Carry}$, $\text{HC} \leftarrow \text{Half Carry}$
SUBC	A,Meml	Subtract with Carry	$A \leftarrow A - \overline{\text{Meml}} + C$, $C \leftarrow \text{Carry}$, $\text{HC} \leftarrow \text{Half Carry}$
AND	A,Meml	Logical AND	$A \leftarrow A \text{ and } \overline{\text{Meml}}$
ANDSZ	A,Imm	Logical AND Immed., Skip if Zero	Skip next if $(A \text{ and } \text{Imm}) = 0$
OR	A,Meml	Logical OR	$A \leftarrow A \text{ or } \text{Meml}$
XOR	A,Meml	Logical EXclusive OR	$A \leftarrow A \text{ xor } \text{Meml}$
IFEQ	MD,Imm	IF EQUAL	Compare MD and Imm, Do next if $\text{MD} = \text{Imm}$
IFEQ	A,Meml	IF EQUAL	Compare A and Meml, Do next if $A = \text{Meml}$
IFNE	A,Meml	IF Not Equal	Compare A and Meml, Do next if $A \neq \text{Meml}$
IFGT	A,Meml	IF Greater Than	Compare A and Meml, Do next if $A > \text{Meml}$
IFBNE	#	If B Not Equal	Do next if lower 4 bits of $B \neq \text{Imm}$
DRSZ	Reg	Decrement Reg., Skip if Zero	$\text{Reg} \leftarrow \text{Reg} - 1$, Skip if $\text{Reg} = 0$
SBIT	#,Mem	Set BIT	1 to bit, Mem (bit = 0 to 7 immediate)
RBIT	#,Mem	Reset BIT	0 to bit, Mem
IFBIT	#,Mem	IF BIT	If bit #,A or Mem is true do next instruction
RPND		Reset PeNDing Flag	Reset Software Interrupt Pending Flag
X	A,Mem	EXchange A with Memory	$A \leftrightarrow \text{Mem}$
X	A,[X]	EXchange A with Memory [X]	$A \leftrightarrow [X]$
LD	A,Meml	LoaD A with Memory	$A \leftarrow \text{Meml}$
LD	A,[X]	LoaD A with Memory [X]	$A \leftarrow [X]$
LD	B,Imm	LoaD B with Immed.	$B \leftarrow \text{Imm}$
LD	Mem,Imm	LoaD Memory Immed.	$\text{Mem} \leftarrow \text{Imm}$
LD	Reg,Imm	LoaD Register Memory Immed.	$\text{Reg} \leftarrow \text{Imm}$
X	A, [B±]	EXchange A with Memory [B]	$A \leftrightarrow [B]$, $(B \leftarrow B \pm 1)$
X	A, [X±]	EXchange A with Memory [X]	$A \leftrightarrow [X]$, $(X \leftarrow X \pm 1)$
LD	A, [B±]	LoaD A with Memory [B]	$A \leftarrow [B]$, $(B \leftarrow B \pm 1)$
LD	A, [X±]	LoaD A with Memory [X]	$A \leftarrow [X]$, $(X \leftarrow X \pm 1)$
LD	[B±],Imm	LoaD Memory [B] Immed.	$[B] \leftarrow \text{Imm}$, $(B \leftarrow B \pm 1)$
CLR	A	CLeaR A	$A \leftarrow 0$
INC	A	INCRe ment A	$A \leftarrow A + 1$
DEC	A	DECRe ment A	$A \leftarrow A - 1$
LAID		Load A InDirect from ROM	$A \leftarrow \text{ROM} (\text{PU}, A)$
DCOR	A	Decimal CORrect A	$A \leftarrow \text{BCD correction of A (follows ADC, SUBC)}$
RRC	A	Rotate A Right thru C	$C \rightarrow A7 \rightarrow \dots \rightarrow A0 \rightarrow C$
RLC	A	Rotate A Left thru C	$C \leftarrow A7 \leftarrow \dots \leftarrow A0 \leftarrow C$, $\text{HC} \leftarrow A0$
SWAP	A	SWAP nibbles of A	$A7 \dots A4 \leftrightarrow A3 \dots A0$
SC		Set C	$C \leftarrow 1$, $\text{HC} \leftarrow 1$
RC		Reset C	$C \leftarrow 0$, $\text{HC} \leftarrow 0$
IFC		IF C	If C is true, do next instruction
IFNC		IF Not C	If C is not true, do next instruction
POP	A	POP the stack into A	$\text{SP} \leftarrow \text{SP} + 1$, $A \leftarrow [\text{SP}]$
PUSH	A	PUSH A onto the stack	$[\text{SP}] \leftarrow A$, $\text{SP} \leftarrow \text{SP} - 1$
VIS		Vector to Interrupt Service Routine	$\text{PU} \leftarrow [\text{VU}]$, $\text{PL} \leftarrow [\text{VL}]$
JMPL	Addr.	Jump absolute Long	$\text{PC} \leftarrow \text{ii}$ (ii = 15 bits, 0 to 32k)
JMP	Addr.	Jump absolute	$\text{PC}9 \dots 0 \leftarrow i$ (i = 12 bits)
JP	Disp.	Jump relative short	$\text{PC} \leftarrow \text{PC} + r$ (r is -31 to +32, except 1)
JSRL	Addr.	Jump SubRoutine Long	$[\text{SP}] \leftarrow \text{PL}$, $[\text{SP}-1] \leftarrow \text{PU}$, $\text{SP}-2$, $\text{PC} \leftarrow \text{ii}$
JSR	Addr.	Jump SubRoutine	$[\text{SP}] \leftarrow \text{PL}$, $[\text{SP}-1] \leftarrow \text{PU}$, $\text{SP}-2$, $\text{PC}9 \dots 0 \leftarrow i$
JSRB	Addr	Jump SubRoutine Boot ROM	$[\text{SP}] \leftarrow \text{PL}$, $[\text{SP}-1] \leftarrow \text{PU}$, $\text{SP}-2$, $\text{PL} \leftarrow \text{Addr}$, $\text{PU} \leftarrow 00$, switch to flash
JID		Jump InDirect	$\text{PL} \leftarrow \text{ROM} (\text{PU}, A)$
RET		RETurn from subroutine	$\text{SP} + 2$, $\text{PL} \leftarrow [\text{SP}]$, $\text{PU} \leftarrow [\text{SP}-1]$
RETSK		RETurn and SKip	$\text{SP} + 2$, $\text{PL} \leftarrow [\text{SP}]$, $\text{PU} \leftarrow [\text{SP}-1]$, skip next instruction
RETI		RETurn from Interrupt	$\text{SP} + 2$, $\text{PL} \leftarrow [\text{SP}]$, $\text{PU} \leftarrow [\text{SP}-1]$, $\text{GIE} \leftarrow 1$
INTR		Generate an Interrupt	$[\text{SP}] \leftarrow \text{PL}$, $[\text{SP}-1] \leftarrow \text{PU}$, $\text{SP}-2$, $\text{PC} \leftarrow \text{OFF}$
NOP		No OPeration	$\text{PC} \leftarrow \text{PC} + 1$