

BRANCH	0	0	c2	c1	c0	o5	o4	o3	o2	o1	o0	Conditional "c" relative jump to (+/-)offset indicated by "o"
HALT	0	0	0	0	0	x	x	x	x	x	x	Stop execution
	0	0	0	0	1	o5	o4	o3	o2	o1	o0	
NOP	0	0	0	1	0	x	x	x	x	x	x	Do nothing, just continue execution
SJMP	0	0	0	1	1	o5	o4	o3	o2	o1	o0	Jump between -31 and +32 bytes relative from PC
BRAZ	0	0	1	0	0	o5	o4	o3	o2	o1	o0	If Zeroflag=1 then jump between -31 and +32 bytes relative from P
BRANZ	0	0	1	0	1	o5	o4	o3	o2	o1	o0	If Zeroflag=0 then jump between -31 and +32 bytes relative from P
BRAC	0	0	1	1	0	o5	o4	o3	o2	o1	o0	If Carry=1 then jump between -31 and +32 bytes relative from PC
BRANC	0	0	1	1	1	o5	o4	o3	o2	o1	o0	If Carry=0 then jump between -31 and +32 bytes relative from PC
LDAI	0	1	0	id7	id6	id5	id4	id3	id2	id1	id0	Load immediate data "id" into A
LDAZP	0	1	1	a7	a6	a5	a4	a3	a2	a1	a0	Load value of RAM in areas 0x00..0xFF into A
STAZP	1	0	0	a7	a6	a5	a4	a3	a2	a1	a0	Store value of A into RAM in areas 0x00..0xFF
ALU	1	0	1	0	op3	op2	op1	op0	r2	r1	r0	Perform the operation "op" on register "r"
CLR	1	0	1	0	0	0	0	0	r2	r1	r0	r = 0x00
SETFF	1	0	1	0	0	0	0	1	r2	r1	r0	r = 0xFF
NOT	1	0	1	0	0	0	1	0	r2	r1	r0	r = r XOR 0xFF
OR	1	0	1	0	0	0	1	1	r2	r1	r0	r = r OR A
AND	1	0	1	0	0	1	0	0	r2	r1	r0	r = r AND A
XOR	1	0	1	0	0	1	0	1	r2	r1	r0	r = r XOR A
INC	1	0	1	0	0	1	1	0	r2	r1	r0	r = r + 1
DEC	1	0	1	0	0	1	1	1	r2	r1	r0	r = r - 1
ADD	1	0	1	0	1	0	0	0	r2	r1	r0	r = r + A
SUB	1	0	1	0	1	0	0	1	r2	r1	r0	r = r - A
ADDC	1	0	1	0	1	0	1	0	r2	r1	r0	r = r + A + carry
SUBC	1	0	1	0	1	0	1	1	r2	r1	r0	r = r - A - carry
LSHIFT	1	0	1	0	1	1	0	0	r2	r1	r0	r = r << 1, lsb=0
RSHIFT	1	0	1	0	1	1	0	1	r2	r1	r0	r = r >> 1, msb=0
LSHIFTC	1	0	1	0	1	1	1	0	r2	r1	r0	r = r << 1, lsb=carry
RSHIFTC	1	0	1	0	1	1	1	1	r2	r1	r0	r = r >> 1, msb=carry
MOVE	1	0	1	1	0	rs2	rs1	rs0	rd2	rd1	rd0	Copy (move) register "rs" to register "rd"
TEST	1	0	1	1	1	b2	b1	b0	r2	r1	r0	Test if bit "b" is set in register "r" and update Z-flag
PEEK	1	1	0	0	0	0	0	0	p3	p2	p1	Load value of i/o port "p" into A
POKE	1	1	0	0	0	0	1	0	p3	p2	p1	Store value of A into i/o port "p"
LDAXY	1	1	0	0	0	1	0	0	0	0	0	Load value of RAM pointed to by X/Y register pair into A-register
STAXY	1	1	0	0	0	1	0	0	0	0	1	Store value of A into RAM pointed to by X/Y register pair
JUMP	1	1	0	0	0	1	0	0	0	1	0	Jump to address pointed to by JH/JL
CALL	1	1	0	0	0	1	0	0	0	1	1	Jump to address pointed to by JH/JL, update PCHOLD with current PC
RETURN	1	1	0	0	0	1	0	0	1	0	0	Return from CALL by loading PC with values from PCHOLD