REGISTER		
0	A	Main working register/accumulator
1	B	General register
2	C	General register
3	D	General register
4	X	General register/High memory pointer
5	Y	General register/Low memory pointer
6	JH	High memory pointer for jumps/calls
7	JL	Low memory pointer for jumps/call

CONDITIONS	
0	HALT
1	
2	NEVER
3	ALWAYS
4	Z-flag set
5	Z-flag clear
6	C-flag set
7	C-flag clear

ALU-OPS			
0	SET 0x00	8	ADD
1	SET 0xFF	9	SUBTRACT
2	INVERT	10	ADD w Carry
3	OR	11	SUB w Carry
4	AND	12	LEFTSHIFT
5	XOR	13	RIGHTSHIFT
6	INCREMENT	14	LEFTSHIFT w Carry
7	DECREMENT	15	RIGHTSHIFT w Carry

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
FREE1	*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 PC
BRANZ	*0*	*0*	*1*	*0*	*1*	*o5*	*o4*	*o3*	*o2*	*o1*	*o0*	If Zeroflag=0 then jump between -31 and +32 bytes relative from PC
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**	**p3**	**p2**	**p1**	**p0**	Load value of i/o port "p" into A
POKE	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**p3**	**p2**	**p1**	**p0**	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
RET	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	Return from CALL by loading PC with values from PCHOLD
	1	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	
CLRC	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	Clear Carry flag
SETC	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	Set Clarry flag
LDPML	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	Load Low byte (bit 0..7) from Program Memry pointed to by XY into A
LDPMH	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	Load High byte (bit 8..10) from Program Memry pointed to by XY into A
STPML	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	Store A into Low byte (bit 0..7) of Program memory pointed to by XY
STPMH	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	Store A into High byte (bit 8..10) of Program memory pointed to by X