

Mnemonic	Byte 1	Byte 2	Byte 3	Pseudocode	Legend:
STR Rs, addr	00000 SSS	AAAA AAAA	AAAA AAAA	mem[addr] = Rs	A - Address
STR Rs, [HL]	00001 SSS			mem[HL] = Rs	CCC - Condition
STR Rs, addr+B	00010 SSS	AAAA AAAA	AAAA AAAA	mem[addr+B] = Rs	DDD - Destination register
STR Rs, [HL]+B	00011 SSS			mem[HL+B] = Rs	I - Immediate
LOD Rd, addr	00100 DDD	AAAA AAAA	AAAA AAAA	Rd = mem[addr]	SSS - Source register
LOD Rd, [HL]	00101 DDD			Rd = mem[HL]	X - Ignored
LOD Rd, addr+B	00110 DDD	AAAA AAAA	AAAA AAAA	Rd = mem[HL+B]	
LOD Rd, [HL]+B	00111 DDD			Rd = mem[HL+B]	Flags (MSB to LSB)
PUSH Rs	01000 SSS			ram[SP] = Rs, SP++	7: Privelege (0/1 - user/supervisor)
POP RD	01001 DDD			Rd = ram[SP], SP--	6: Interrupts (0/1 - disabled/enabled)
JMP addr	0101L CCC	AAAA AAAA	AAAA AAAA	IF (C) PC = addr	5: Overflow
JMP [HL]	0110L CCC			IF (C) PC = HL	4: Sign
INT id	0111 CCC 0	IIII IIII		IF (C) PUSH PC, FLAGS, JMP INTVEC	3: Auxiliary carry
IRET	0111 CCC 1			IF (C) POP FLAGS, PC	2: Parity
CALL addr	1000 CCC 0	AAAA AAAA	AAAA AAAA	IF (C) PUSH PC+3, JMP addr	1: Carry
CALL [HL]	1000 CCC 1			IF (C) PUSH PC+3, JMP HL	0: Zero
RET	10010 CCC			IF (C) POP PC	
ADD Rd, Rs, Rs	10011 DDD	SSS sss 00		Rd = Rs + Rs	Memory writes to SSS == 0 are equivalent
ADC Rd, Rs, Rs	10011 DDD	SSS sss 01		Rd = Rs + Rs + C	to NOP
SUB Rd, Rs, Rs	10011 DDD	SSS sss 10		Rd = Rs - Rs	
SBB Rd, Rs, Rs	10011 DDD	SSS sss 11		Rd = Rs - Rs - B	
AND Rd, Rs, Rs	10100 DDD	SSS sss 00		Rd = Rs && Rs	
NOR Rd, Rs, Rs	10100 DDD	SSS sss 01		Rd = !(Rs    Rs)	
XOR Rd, Rs, Rs	10100 DDD	SSS sss 10		Rd = Rs ^ Rs	
NEG Rd	10100 DDD	XXX 000 11		Rd = -Rd	
SHL Rd, Amnt	10100 DDD	AAA 001 11		Rd <<= A	
SHR Rd, Amnt	10100 DDD	AAA 010 11		Rd >>= A	
ROL Rd, Amnt	10100 DDD	AAA 011 11		Rd = (Rd << A)    (Rd >> (8 - A))	
ROR Rd, Amnt	10100 DDD	AAA 100 11		Rd = (Rd >> A)    (Rd << (8 - A))	
SAL Rd, Amnt	10100 DDD	AAA 101 11		signed Rd <<= A	
SAR Rd, Amnt	10100 DDD	AAA 110 11		signed Rd >>= A	
ABS Rd	10100 DDD	XXX 111 11		Rd = Abs(Rd)	
ADI Rd, Rd + Imm8	10101 DDD	IIII IIII		Rd += Imm8	
LDI Rd, Imm8	10110 DDD	IIII IIII		Rd = Imm8	
SBI Rd, Rd + Imm8	10111 DDD	IIII IIII		Rd -= Imm8	
INC Rd	11000 DDD			Rd++	
DEC Rd	11001 DDD			Rd--	
WFI	11010 000			while (IRQ  NMI) {sleep}	
SET	11010 001	IIII IIII		flags = flags    imm8	
CLEAR	11010 010	IIII IIII		flags = flags && !imm8	
MASK	11010 011	IIII IIII		mask = imm8	
CPR	11010 100			HL = PC	
CSR	11010 101			HL = SP	
CBR	11010 110			HL = BP	
LOOP	11010 111	AAAA AAAA	AAAA AAAA	while (R3 != 0) { jmp mem[addr]; R3-- }	
MUL Rs, Rs	11011 CCC	SSS SSS 00		if (C) HL = Rs * Rs	
IMUL Rs, Rs	11011 CCC	SSS SSS 01		if (C) HL = Rs * Rs // signed	
DIV Rs, Rs	11011 CCC	SSS SSS 10		if (C) Hi = Rs / Rs; Lo = Rs % Rs	
IDIV Rs, Rs	11011 CCC	SSS SSS 11		if (C) Hi = Rs / Rs; Lo = Rs % Rs // signed	
MLI Rs, Imm8	11100 SSS	IIII IIII		HL = Rs * imm8	
IMLI Rs, Imm8	11101 SSS	IIII IIII		HL = Rs * imm8 // signed	
DVI Rs, Imm8	11110 SSS	IIII IIII		Hi = Rs / imm8; Lo = Rs % imm8	
IDVI Rs, Imm8	11111 SSS	IIII IIII		Hi = Rs / imm8; Lo = Rs % imm8 // signed	SSS - 000 - A; 001 - B etc...
Halt	11111111			else if (SSS == 7) quit()	110 - L; 111 - not idvi, HALT