

Gameboy CPU (LR35902) instruction set

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	x
0x	NOP	LD BC,d16	LD (BC),A	INC BC	INC B	DEC B	LD B,d8	RLCA	LD (a16),SP	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DE
	1 4	3 12	1 8	1 8	1 4	1 4	2 8	1 4	3 20	1 8	1 8	1 8	1 4	1
	----	----	----	----	Z 0 H -	Z 1 H -	----	0 0 0 C	----	- 0 H C	----	----	Z 0 H -	Z 1
1x	STOP 0	LD DE,d16	LD (DE),A	INC DE	INC D	DEC D	LD D,d8	RLA	JR r8	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DE
	2 4	3 12	1 8	1 8	1 4	1 4	2 8	1 4	2 12	1 8	1 8	1 8	1 4	1
	----	----	----	----	Z 0 H -	Z 1 H -	----	0 0 0 C	----	- 0 H C	----	----	Z 0 H -	Z 1
2x	JR NZ,r8	LD HL,d16	LD (HL+),A	INC HL	INC H	DEC H	LD H,d8	DAA	JR Z,r8	ADD HL,HL	LD A,(HL+)	DEC HL	INC L	DE
	2 12/8	3 12	1 8	1 8	1 4	1 4	2 8	1 4	2 12/8	1 8	1 8	1 8	1 4	1
	----	----	----	----	Z 0 H -	Z 1 H -	----	Z - 0 C	----	- 0 H C	----	----	Z 0 H -	Z 1
3x	JR NC,r8	LD SP,d16	LD (HL-),A	INC SP	INC (HL)	DEC (HL)	LD (HL),d8	SCF	JR C,r8	ADD HL,SP	LD A,(HL-)	DEC SP	INC A	DE
	2 12/8	3 12	1 8	1 8	1 12	1 12	2 12	1 4	2 12/8	1 8	1 8	1 8	1 4	1
	----	----	----	----	Z 0 H -	Z 1 H -	----	- 0 0 1	----	- 0 H C	----	----	Z 0 H -	Z 1
4x	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
5x	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
6x	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
7x	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD
	1 8	1 8	1 8	1 8	1 8	1 8	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
8x	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0 H C	Z 0
9x	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1
Ax	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOI
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 1 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0
Bx	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CF
	1 4	1 4	1 4	1 4	1 4	1 4	1 8	1 4	1 4	1 4	1 4	1 4	1 4	1
	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1 H C	Z 1
Cx	RET NZ	POP BC	JP NZ,a16	JP a16	CALL NZ,a16	PUSH BC	ADD A,d8	RST 00H	RET Z	RET	JP Z,a16	PREFIX CB	CALL Z,a16	CALL
	1 20/8	1 12	3 16/12	3 16	3 24/12	1 16	2 8	1 16	1 20/8	1 16	3 16/12	1 4	3 24/12	3
	----	----	----	----	----	----	Z 0 H C	----	----	----	----	----	----	--
Dx	RET NC	POP DE	JP NC,a16		CALL NC,a16	PUSH DE	SUB d8	RST 10H	RET C	RETI	JP C,a16		CALL C,a16	
	1 20/8	1 12	3 16/12		3 24/12	1 16	2 8	1 16	1 20/8	1 16	3 16/12		3 24/12	
	----	----	----		----	----	Z 1 H C	----	----	----	----		----	
Ex	LDH (a8),A	POP HL	LD (C),A			PUSH HL	AND d8	RST 20H	ADD SP,r8	JP (HL)	LD (a16),A			
	2 12	1 12	2 8			1 16	2 8	1 16	2 16	1 4	3 16			
	----	----	----			Z 0 1 0	----	----	0 0 H C	----	----			
Fx	LDH A,(a8)	POP AF	LD A,(C)	DI		PUSH AF	OR d8	RST 30H	LD HL,SP+r8	LD SP,HL	LD A,(a16)	EI		
	2 12	1 12	2 8	1 4		1 16	2 8	1 16	2 12	1 8	3 16	1 4		
	----	Z N H C	----	----		----	Z 0 0 0	----	0 0 H C	----	----	----		

Prefix CB

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xi
0x	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0
1x	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0
2x	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0
3x	SWAP B	SWAP C	SWAP D	SWAP E	SWAP H	SWAP L	SWAP (HL)	SWAP A	SRL B	SRL C	SRL D	SRL E	SRL H	SRL
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 0	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0 0 C	Z 0
4x	BIT 0,B	BIT 0,C	BIT 0,D	BIT 0,E	BIT 0,H	BIT 0,L	BIT 0,(HL)	BIT 0,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0
5x	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0
6x	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0
7x	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0 1 -	Z 0
8x	RES 0,B	RES 0,C	RES 0,D	RES 0,E	RES 0,H	RES 0,L	RES 0,(HL)	RES 0,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
9x	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
Ax	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
Bx	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
Cx	SET 0,B	SET 0,C	SET 0,D	SET 0,E	SET 0,H	SET 0,L	SET 0,(HL)	SET 0,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--
Dx	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
	----	----	----	----	----	----	----	----	----	----	----	----	----	--

Ex	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2
Fx	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET
	2 8	2 8	2 8	2 8	2 8	2 8	2 16	2 8	2 8	2 8	2 8	2 8	2 8	2

Misc/control instructions
Jumps/calls
8bit load/store/move instructions
16bit load/store/move instructions
8bit arithmetic/logical instructions
16bit arithmetic/logical instructions
8bit rotations/shifts and bit instructions

Length in bytes →

INS reg

2 8

Z N H C

← Instruction mnemonic

← Duration in cycles

← Flags affected

Duration of conditional calls and ret is taken or not. This is indicated by The higher number (on the left side of instruction when action is taken, the side of "/") means duration of instru

Instruction **STOP** has according to manuals opcode **10 00** and thus is 2 bytes long. Anyhow it seems there is no reason for it so some assemblers code it simply as one byte inst: Flags affected are always shown in **Z H N C** order. If flag is marked by "0" it means it is reset after the instruction. If it is marked by "1" it is set. If it is marked by "." marked by "Z", "N", "H" or "C" corresponding flag is affected as expected by its function.

- d8 means immediate 8 bit data
- d16 means immediate 16 bit data
- a8 means 8 bit unsigned data, which are added to \$FF00 in certain instructions (replacement for missing **IN** and **OUT** instructions)
- a16 means 16 bit address
- r8 means 8 bit signed data, which are added to program counter

LD A, (C) has alternative mnemonic **LD A, (\$FF00+C)**
LD C, (A) has alternative mnemonic **LD (\$FF00+C), A**
LDH A, (a8) has alternative mnemonic **LD A, (\$FF00+a8)**
LDH (a8), A has alternative mnemonic **LD (\$FF00+a8), A**
LD A, (HL+) has alternative mnemonic **LD A, (HLI)** or **LDI A, (HL)**
LD (HL+), A has alternative mnemonic **LD (HLI), A** or **LDI (HL), A**
LD A, (HL-) has alternative mnemonic **LD A, (HLD)** or **LDD A, (HL)**
LD (HL-), A has alternative mnemonic **LD (HLD), A** or **LDD (HL), A**
LD HL, SP+r8 has alternative mnemonic **LDHL SP, r8**

Registers

15 ... 8	7 ... 0
A (accumulator)	F (flags)
B	C
D	E
H	L

15 ... 0
SP (stack pointer)
PC (program counter)

Flag register (F) bits:

7	6	5	4	3	2	1	0
Z	N	H	C	0	0	0	0

- Z** - Zero Flag
- N** - Subtract Flag
- H** - Half Carry Flag
- C** - Carry Flag
- 0** - Not used, always zero