Gameboy CPU (LR35902) instruction set

	х0	x1	x2	х3	x4	x5	х6	x7	x8	x9	xΑ	хB	xC	хD	хE	xF
	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	DEC C	LD C,d8	RRCA
0x	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 4	2 8	1 4
					Z 0 H -	Z 1 H -		000C		- 0 H C			Z 0 H -	Z 1 H -		000C
	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	DEC E	LD E,d8	RRA
1x	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 4	2 8	1 4
					Z 0 H -	Z 1 H -		000C		- 0 H C			Z 0 H -	Z 1 H -		000C
	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	DEC L	LD L,d8	CPL
2x	2 12/8	3 12	1 8	1 8	1 4 Z 0 H -	1 4	28	1 4 Z - 0 C	2 12/8	1 8 - 0 H C	1 8	1 8	1 4	1 4 Z 1 H -	2 8	1 4 - 1 1 -
				INC SP	INC (HL)	Z 1 H -		SCF		ADD HL,SP		DEC SP	Z O H -	DEC A		- 1 1 -
3x	JR NC,r8 2 12/8	LD SP,d16 3 12	LD (HL-),A 1 8	1 8	1 12	DEC (HL) 1 12	LD (HL),d8 2 12	1 4	JR C,r8 2 12/8	1 8	LD A,(HL-) 1 8	1 8	INC A 1 4	1 4	LD A,d8 2 8	1 4
эх	2 12/8	3 12	1 8		Z 0 H -	Z 1 H -	2 12	-001	2 12/8	- 0 H C			Z 0 H -	Z 1 H -	2 8 	- 0 0 C
	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 C,L	LD C,(HL)	LD C,A
4x	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 4	1 8	1 4
47																
	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 E,L	LD E,(HL)	LD E,A
5x	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 4	1 8	1 4
	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 L,L	LD L,(HL)	LD L,A
6x	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 4	1 8	1 4
	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 A,L	LD A,(HL)	LD A,A
7x	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 4	1 8	1 4
	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 A,L	ADC A,(HL)	ADC A,A
8x	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 4	1 8	1 4
	Z 0 H C	Z Ø H C	ZOHC	Z 0 H C	Z O H C	Z 0 H C	ZOHC	Z 0 H C	ZOHC	ZOHC	Z 0 H C	ZOHC	ZOHC	Z 0 H C	Z 0 H C	Z 0 H C
	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 A,L	SBC A,(HL)	SBC A,A
9x	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 4	1 8	1 4
	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 H C	Z 1 H C	Z 1 H C
Ax	AND B 1 4	AND C 1 4	AND D 1 4	AND E 1 4	AND H 1 4	AND L 1 4	AND (HL) 1 8	AND A 1 4	XOR B 1 4	XOR C 1 4	XOR D 1 4	XOR E 1 4	XOR H 1 4	XOR L 1 4	XOR (HL) 1 8	XOR A
AX	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 0 0	Z 0 0 0	1 4 Z 0 0 0
	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	CP L	CP (HL)	CP A
Bx	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 4	1 8	1 4
DX	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 H C	Z 1 H C	Z 1 H C
	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 a16	ADC A,d8	RST 08H
Cx	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 24	2 8	1 16
							Z 0 H C								Z 0 H C	
	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		SBC A,d8	RST 18H
Dx	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		2 8	1 16
							Z 1 H C								Z 1 H C	
	LDH (a8),A	POP HL	LD (C),A			PUSH HL	AND d8	RST 20H	ADD SP,r8	JP (HL)	LD (a16),A				XOR d8	RST 28H
Ex	2 12	1 12	2 8			1 16	2 8	1 16	2 16	1 4	3 16				2 8	1 16
							Z 0 1 0		0 0 H C						Z 0 0 0	
	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			CP d8	RST 38H
Fx	2 12	1 12	2 8	1 4		1 16	2 8	1 16	2 12	1 8	3 16	1 4			2 8	1 16
		ZNHC					Z 0 0 0		0 0 H C						Z 1 H C	

Prefix CB

	x0	x1	x2	x3	x4	x5	х6	х7	x8	x9	хA	хB	xC	хD	хE	хF
	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 L	RRC (HL)	RRC A
0x	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 8	2 16	2 8
	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 0 C	Z 0 0 C	Z 0 0 C
	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 L	RR (HL)	RR A
1x	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 8	2 16	2 8
	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 0 C	Z 0 0 C	Z 0 0 C
	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 L	SRA (HL)	SRA A
2x	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 8	2 16	2 8
	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 0 0	Z 0 0 0	Z 0 0 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 L	SRL (HL)	SRL A
	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 8	2 16	2 8

	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 0 C	Z 0 0 C	Z 0 0 C
	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 1,L	BIT 1,(HL)	BIT 1,A
4x	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 8	2 16	2 8
	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 1 -	Z 0 1 -	Z 0 1 -
	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 3,L	BIT 3,(HL)	BIT 3,A
5x	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 8	2 16	2 8
	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 1 -	Z 0 1 -	Z 0 1 -
	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 5,L	BIT 5,(HL)	BIT 5,A
6x	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 8	2 16	2 8
	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 1 -	Z 0 1 -	Z 0 1 -
	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 7,L	BIT 7,(HL)	BIT 7,A
7x	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 8	2 16	2 8
	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 1 -	Z 0 1 -	Z 0 1 -
	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 1,L	RES 1,(HL)	RES 1,A
8x	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 8	2 16	2 8
	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 3,L	RES 3,(HL)	RES 3,A
9x	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 8	2 16	2 8
	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 5,L	RES 5,(HL)	RES 5,A
Ax	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 8	2 16	2 8
	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 7,L	RES 7,(HL)	RES 7,A
Bx	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 8	2 16	2 8
Сх	SET 0,B 2 8	SET 0,C 2 8	SET 0,D 2 8	SET 0,E 2 8	SET 0,H 2 8	SET 0,L 2 8	SET 0,(HL) 2 16	SET 0,A 2 8	SET 1,B 2 8	SET 1,C 2 8	SET 1,D 2 8	SET 1,E 2 8	SET 1,H 2 8	SET 1,L 2 8	SET 1,(HL) 2 16	SET 1,A 2 8
CX	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 8	2 16	2 8
	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 3,L	SET 3,(HL)	SET 3,A
Dx	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 8	2 16	2 8
DX	2 0	2 0	2 0	2 0	2 0	2 0	2 16	2 0	2 0	2 0	2 0	2 0	2 0	2 0	2 16	2 0
	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 5,L	SET 5,(HL)	SET 5,A
Fx	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 8	2 16	2 8
	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 7,L	SET 7,(HL)	SET 7,A
Fx	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 8	2 16	2 8

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 \rightarrow INS reg
2 8
2 N H C

The purpose of the second in Exercision Exercision Exercision Exercises (Continuous Exercises Appendix A

Duration of conditional calls and returns is different when action is taken or not. This is indicated by two numbers separated by "/". The higher number (on the left side of "/") means duration of instruction when action is taken, the lower number (on the right side of "/") means duration of instruction when action is not taken.

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 instruction 10.

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 "-" it is not changed. If it is 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

imply as one byte instruction 10.

Registers

15 8	7 0
A (accumulator)	F (flags)
В	С
D	Е
Н	L

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

Flag register (F) bits:

7	6	5	4	3	2	1	0
Z	N	Н	С	0	0	0	0

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