Instructions

A Guide to the Instruction Schema,
Implementation of Instruction Logic, both of
Individual Instructions and Shared Logic, and the
Operations of the Individual Instructions

Miscellaneous

Machine Control, Program Flow,
Stack Control and Accumulator
Arithmetic and Logic Operations

Overview

Common Structure: 0 – Row – Operation

This logic block contains all the machine control, program flow, double register and accumulator operations.

Operations:

- [0] Machine Control
- [1] Jump
- [2] Subroutine
- [3] Return
- [4] Push
- [5] Pop
- [6] Double Register Increment/Decrement
- [7] Accumulator Operations

Registers:

- [0] Xh Register
- [1] Xl Register
- [2] Yh Register
- [3] Yl Register
- [4] H Register
- [5] L Register
- [6] Accumulator
- [7] Stack Pointer

Machine Control Rows:

- [0] No Instruction
- [1] Clear Carry
- [2] Clear Negative
- [3] Clear Zero
- [4] Clear Interrupt
- [5] Clear Flags
- [6] Extended Set
- [7] Halt

Condition Rows:

[0] Unconditional
[1] Carry
[2] Negative
[3] Zero
[4] Not Carry
[5] Not Negative
[6] Not Zero
[7] Interrupt

Double Register Increment/Decrement Rows:

[0] Increment
[1] Decrement
[2] Increment
[3] Decrement
[4] Increment
[5] Decrement

Accumulator Operations:

[0] Increment
[1] Decrement
[2] Rotate Right
[3] Rotate Left
[4] Right Shift
[5] Left Shift
[6] NOT

No Instruction

Operation: N/A
Structure: N/A

No instruction.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	NIN	000	1	2	-	-	-	-

Clear Carry Flag

Operation: C ← 0
Structure: N/A

Resets the Carry Flag to zero.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	CLC	010	1	2	0	-	-	-

Clear Negative Flag

Operation: N ← 0
Structure: N/A

Resets the Negative Flag to zero.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	CLN	020	1	2	-	0	-	-

Clear Zero Flag

Operation: Z ← 0
Structure: N/A

Resets the Zero Flag to zero.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	CLZ	030	1	2	-	-	0	-

Clear Interrupt Flag

Operation: I ← 0
Structure: N/A

Resets the Interrupt Flag to zero.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	CLI	040	1	2	-	-	-	0

Reset Flag Register

Operation: C, N, Z, I ← 0
Structure: N/A

Resets every Flag in F to zero.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	CLF	050	1	2	0	0	0	0

Extended Instruction Mode

Operation: N/A
Structure: N/A

Enables the Extended Instruction Block for the next instruction. Extended Instruction Mode clears after the next instruction finishes.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	EXT	060	1	2	-	-	-	-

Halt Clock

Operation: N/A
Structure: N/A

Halts the Clock Module. May be continued from front panel.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	HLT	070	1	2	-	-	-	-

Jump

Operation: PC ← $hhll if Condition
Structure: 0 – Condition – 1

Jumps to the address specified after the instruction if the condition is met.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	JMP u, $hhll	001	3	4	-	-	-	-
Absolute	JMP c, $hhll	011	3	4	?	-	-	-
Absolute	JMP n, $hhll	021	3	4	-	?	-	-
Absolute	JMP z, $hhll	031	3	4	-	-	?	-
Absolute	JMP nc, $hhll	041	3	4	?	-	-	-
Absolute	JMP nn, $hhll	051	3	4	-	?	-	-
Absolute	JMP nz, $hhll	061	3	4	-	-	?	-
Absolute	JMP i, $hhll	071	3	4	-	-	-	?

Subroutine

Operation: (Stack ← PC, SP ← SP + 2, PC ← $hhll) if Condition

Structure: 0 – Condition – 2

Jumps to the address specified after the instruction if the condition is met, pushes previous PC value to the Stack.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	JSR u, $hhll	002	3	5	-	-	-	-
Absolute	JSR c, $hhll	012	3	5	?	-	-	-
Absolute	JSR n, $hhll	022	3	5	-	?	-	-
Absolute	JSR z, $hhll	032	3	5	-	-	?	-
Absolute	JSR nc, $hhll	042	3	5	?	-	-	-
Absolute	JSR nn, $hhll	052	3	5	-	?	-	-
Absolute	JSR nz, $hhll	062	3	5	-	-	?	-
Absolute	JSR i, $hhll	072	3	5	-	-	-	?

Return

Operation: (SP ← SP - 2, PC ← Stack) if Condition
Structure: 0 – Condition – 3

Pops the topmost value from the Stack into the PC.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	RTN u	003	1	6	-	-	-	-
Implied	RTN c	013	1	6	?	-	-	-
Implied	RTN n	023	1	6	-	?	-	-
Implied	RTN z	033	1	6	-	-	?	-
Implied	RTN nc	043	1	6	?	-	-	-
Implied	RTN nn	053	1	6	-	?	-	-
Implied	RTN nz	063	1	6	-	-	?	-

Push

Operation: Stack ← Register, SP ← SP + 1
Structure: 0 – Register – 4

Pushes the register specified in the instruction onto the Stack.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	PSH Xh	004	1	3	-	-	-	-
Register	PSH Xl	014	1	3	-	-	-	-
Register	PSH Yh	024	1	3	-	-	-	-
Register	PSH Yl	034	1	3	-	-	-	-
Register	PSH H	044	1	3	-	-	-	-
Register	PSH L	054	1	3	-	-	-	-
Register	PSH Acc	064	1	3	-	-	-	-

Pop

Operation: SP ← SP – 1, Stack ← Register
Structure: 0 – Register – 5

Pops the topmost value from the Stack into the register specified in the instruction.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	POP Xh	005	1	3	-	-	-	-
Register	POP Xl	015	1	3	-	-	-	-
Register	POP Yh	025	1	3	-	-	-	-
Register	POP Yl	035	1	3	-	-	-	-
Register	POP H	045	1	3	-	-	-	-
Register	POP L	055	1	3	-	-	-	-
Register	POP Acc	065	1	3	-	-	-	-

16 Bit Increment

Operation: Double Register ← Double Register + 1
Structure: 0 – Double Register – 6

Increments the pseudo-16-bit register specified by the instruction.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	INC Xhl	006	1	2	-	-	-	-
Double Register	INC Yhl	026	1	2	-	-	-	-
Double Register	INC HL	046	1	2	-	-	-	-

16 Bit Decrement

Operation: Double Register ← Double Register - 1
Structure: 0 – Double Register – 6

Decrements the pseudo-16-bit register specified by the instruction.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	DEC Xhl	016	1	2	-	-	-	-
Double Register	DEC Yhl	036	1	2	-	-	-	-
Double Register	DEC HL	056	1	2	-	-	-	-

Increment Accumulator

Operation: Acc, C, N, Z ← Acc + 1
Structure: N/A

Increments the Accumulator.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	INC	007	1	2	✓	✓	✓	-

Decrement Accumulator

Operation: Acc, C, N, Z ← Acc - 1
Structure: N/A

Decrements the Accumulator.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	DEC	017	1	2	✓	✓	✓	-

Rotate Accumulator Right

Operation: Acc, C ← Acc ÷ 2 + C
Structure: N/A

Rotates the Accumulator to the right.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	ROR	027	1	2	✓	✓	✓	-

Rotate Accumulator Left

Operation: C, Acc ← Acc × 2 + C
Structure: N/A

Rotates the Accumulator to the left.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	ROL	037	1	2	✓	✓	✓	-

Shift Accumulator Right

Operation: Acc, C ← Acc ÷ 2
Structure: N/A

Shifts the Accumulator to the right.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	RSH	047	1	2	✓	✓	✓	-

Shift Accumulator Left

Operation: Acc, C ← Acc × 2
Structure: N/A

Shifts the Accumulator to the left.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	DEC	057	1	2	✓	✓	✓	-

Bitwise Invert

Operation: Acc ← NOT Acc
Structure: N/A

Inverts the Accumulator.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Implied	NOT	067	1	2	✓	✓	✓	-

Arithmetic and Logic

Addition, Subtraction,
Both with and without Carry,
Compare, and Logic Operations

Overview

Common Structure: 1 – Operation – Operand
This logic block contains solely logic and arithmetic operations. Every operation changes the Carry, Negative and Zero flags.

Operations:

- [0] Add
- [1] Add with Carry
- [2] Subtract
- [3] Subtract with Carry
- [4] Compare
- [5] AND
- [6] OR
- [7] XOR

Operands:

- [0] Absolute Address
- [1] Zero Page Address
- [2] HL Indirect
- [3] Immediate
- [4] Xh Register
- [5] Xl Register
- [6] Yh Register
- [7] Yl Register

Add

Operation: Acc, C, N, Z ← Acc + Operand
Structure: 1 – 0 – Operand

Adds the value in the Accumulator to the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	ADD $hhll	100	3	4	✓	✓	✓	-
Zero Page	ADD $zz	101	2	3	✓	✓	✓	-
HL Indirect	ADD (HL)	102	1	2	✓	✓	✓	-
Immediate	ADD #$nn	103	2	3	✓	✓	✓	-
Register	ADD Xh	104	1	2	✓	✓	✓	-
Register	ADD Xl	105	1	2	✓	✓	✓	-
Register	ADD Yh	106	1	2	✓	✓	✓	-
Register	ADD Yl	107	1	2	✓	✓	✓	-

Add with Carry

Operation: Acc, C, N, Z ← Acc + Operand + C
Structure: 1 – 1 – Operand

Adds the value in the Accumulator to the Operand with Carry In.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	ADC $hhll	110	3	4	✓	✓	✓	-
Zero Page	ADC $zz	111	2	3	✓	✓	✓	-
HL Indirect	ADC (HL)	112	1	2	✓	✓	✓	-
Immediate	ADC #$nn	113	2	3	✓	✓	✓	-
Register	ADC Xh	114	1	2	✓	✓	✓	-
Register	ADC Xl	115	1	2	✓	✓	✓	-
Register	ADC Yh	116	1	2	✓	✓	✓	-
Register	ADC Yl	117	1	2	✓	✓	✓	-

Subtract

Operation: Acc, C, N, Z ← Acc - Operand
Structure: 1 – 2 – Operand

Subtracts the value in the Accumulator to the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	SUB $hhll	120	3	4	✓	✓	✓	-
Zero Page	SUB $zz	121	2	3	✓	✓	✓	-
HL Indirect	SUB (HL)	122	1	2	✓	✓	✓	-
Immediate	SUB #$nn	123	2	3	✓	✓	✓	-
Register	SUB Xh	124	1	2	✓	✓	✓	-
Register	SUB Xl	125	1	2	✓	✓	✓	-
Register	SUB Yh	126	1	2	✓	✓	✓	-
Register	SUB Yl	127	1	2	✓	✓	✓	-

Subtract with Borrow

Operation: Acc, C, N, Z ← Acc - Operand - C
Structure: 1 – 3 – Operand

Subtracts the value in the Accumulator to the Operand with Carry In.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	SBB $hhll	130	3	4	✓	✓	✓	-
Zero Page	SBB $zz	131	2	3	✓	✓	✓	-
HL Indirect	SBB (HL)	132	1	2	✓	✓	✓	-
Immediate	SBB #$nn	133	2	3	✓	✓	✓	-
Register	SBB Xh	134	1	2	✓	✓	✓	-
Register	SBB Xl	135	1	2	✓	✓	✓	-
Register	SBB Yh	136	1	2	✓	✓	✓	-
Register	SBB Yl	137	1	2	✓	✓	✓	-

Compare

Operation: C, N, Z ← Acc - Operand
Structure: 1 – 4 – Operand

Subtracts the value in the Accumulator to the Operand, but does not save the difference.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	CMP $hhll	140	3	4	✓	✓	✓	-
Zero Page	CMP $zz	141	2	3	✓	✓	✓	-
HL Indirect	CMP (HL)	142	1	2	✓	✓	✓	-
Immediate	CMP #$nn	143	2	3	✓	✓	✓	-
Register	CMP Xh	144	1	2	✓	✓	✓	-
Register	CMP Xl	145	1	2	✓	✓	✓	-
Register	CMP Yh	146	1	2	✓	✓	✓	-
Register	CMP Yl	147	1	2	✓	✓	✓	-

AND Operation

Operation: Acc, C, N, Z ← Acc AND Operand
Structure: 1 – 5 – Operand

Performs the AND operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	AND $hhll	150	3	4	✓	✓	✓	-
Zero Page	AND $zz	151	2	3	✓	✓	✓	-
HL Indirect	AND (HL)	152	1	2	✓	✓	✓	-
Immediate	AND #$nn	153	2	3	✓	✓	✓	-
Register	AND Xh	154	1	2	✓	✓	✓	-
Register	AND Xl	155	1	2	✓	✓	✓	-
Register	AND Yh	156	1	2	✓	✓	✓	-
Register	AND Yl	157	1	2	✓	✓	✓	-

OR Operation

Operation: Acc, C, N, Z ← Acc OR Operand
Structure: 1 – 6 – Operand

Performs the OR operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	OR $hhll	160	3	4	✓	✓	✓	-
Zero Page	OR $zz	161	2	3	✓	✓	✓	-
HL Indirect	OR (HL)	162	1	2	✓	✓	✓	-
Immediate	OR #$nn	163	2	3	✓	✓	✓	-
Register	OR Xh	164	1	2	✓	✓	✓	-
Register	OR Xl	165	1	2	✓	✓	✓	-
Register	OR Yh	166	1	2	✓	✓	✓	-
Register	OR Yl	167	1	2	✓	✓	✓	-

XOR Operation

Operation: Acc, C, N, Z ← (Acc OR Operand) AND NOT (Acc AND Operand)
Structure: 1 – 7 – Operand

Performs the XOR operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	XOR $hhll	170	3	4	✓	✓	✓	-
Zero Page	XOR $zz	171	2	3	✓	✓	✓	-
HL Indirect	XOR (HL)	172	1	2	✓	✓	✓	-
Immediate	XOR #$nn	173	2	3	✓	✓	✓	-
Register	XOR Xh	174	1	2	✓	✓	✓	-
Register	XOR Xl	175	1	2	✓	✓	✓	-
Register	XOR Yh	176	1	2	✓	✓	✓	-
Register	XOR Yl	177	1	2	✓	✓	✓	-

Register-Memory Interface

Load and Store Operations

Overview

Common Structure for Load: 2 – Destination – Source
Common Structure for Store: 2 – Source – Destination
This logic block contains the Load and Store instructions.

Load Sources:

- [0] Absolute Address
- [1] Zero Page Address
- [2] HL Indirect
- [3] Immediate

Store Destinations:

- [0] Absolute Address
- [1] Zero Page Address
- [2] HL Indirect

Load Destinations/Store Sources:

- [0] Xh Register
- [1] Xl Register
- [2] Yh Register
- [3] Yl Register
- [4] H Register
- [5] L Register
- [6] Acc
- [7] SP

Load Xh

Operation: Xh ← Memory
Structure: 2 – 0 – Load Source

Loads a value from Memory into Register Xh.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD Xh, $hhll	200	3	4	-	-	-	-
Zero Page	LD Xh, $zz	201	2	3	-	-	-	-
HL Indirect	LD Xh, (HL)	202	1	2	-	-	-	-
Immediate	LD Xh, #$nn	203	2	3	-	-	-	-

Load Xl

Operation: Xl ← Memory
Structure: 2 – 1 – Load Source

Loads a value from Memory into Register Xl.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD Xl, $hhll	210	3	4	-	-	-	-
Zero Page	LD Xl, $zz	211	2	3	-	-	-	-
HL Indirect	LD Xl, (HL)	212	1	2	-	-	-	-
Immediate	LD Xl, #$nn	213	2	3	-	-	-	-

Load Yh

Operation: Yh ← Memory
Structure: 2 – 2 – Load Source

Loads a value from Memory into Register Yh.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD Yh, $hhll	220	3	4	-	-	-	-
Zero Page	LD Yh, $zz	221	2	3	-	-	-	-
HL Indirect	LD Yh, (HL)	222	1	2	-	-	-	-
Immediate	LD Yh, #$nn	223	2	3	-	-	-	-

Load Yl

Operation: Yl ← Memory
Structure: 2 – 3 – Load Source

Loads a value from Memory into Register Yl.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD Yl, $hhll	230	3	4	-	-	-	-
Zero Page	LD Yl, $zz	231	2	3	-	-	-	-
HL Indirect	LD Yl, (HL)	232	1	2	-	-	-	-
Immediate	LD Yl, #$nn	233	2	3	-	-	-	-

Load H

Operation: H ← Memory
Structure: 2 – 4 – Load Source

Loads a value from Memory into Register H.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD H, $hhll	240	3	4	-	-	-	-
Zero Page	LD H, $zz	241	2	3	-	-	-	-
Immediate	LD H, #$nn	243	2	3	-	-	-	-

Load L

Operation: L ← Memory
Structure: 2 – 5 – Load Source

Loads a value from Memory into Register L.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD L, $hhll	250	3	4	-	-	-	-
Zero Page	LD L, $zz	251	2	3	-	-	-	-
Immediate	LD L, #$nn	253	2	3	-	-	-	-

Load Accumulator

Operation: Acc ← Memory
Structure: 2 – 6 – Load Source

Loads a value from Memory into the Accumulator.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD Acc, $hhll	260	3	4	-	-	-	-
Zero Page	LD Acc, $zz	261	2	3	-	-	-	-
HL Indirect	LD Acc, (HL)	262	1	2	-	-	-	-
Immediate	LD Acc, #$nn	263	2	3	-	-	-	-

Load Stack Pointer

Operation: SP ← Memory
Structure: 2 – 7 – Load Source

Loads a value from Memory into the Stack Pointer.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	LD SP, $hhll	270	3	4	-	-	-	-
Zero Page	LD SP, $zz	271	2	3	-	-	-	-
HL Indirect	LD SP, (HL)	272	1	2	-	-	-	-
Immediate	LD SP, #$nn	273	2	3	-	-	-	-

Store Xh

Operation: Memory ← Xh
Structure: 2 – 0 – Store Destination

Stores the value in Register Xh to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR Xh, $hhll	204	3	4	-	-	-	-
Zero Page	STR Xh, $zz	205	2	3	-	-	-	-
HL Indirect	STR Xh, (HL)	206	1	2	-	-	-	-

Store Xl

Operation: Memory ← Xl
Structure: 2 – 1 – Store Destination

Stores the value in Register Xl to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR Xl, $hhll	214	3	4	-	-	-	-
Zero Page	STR Xl, $zz	215	2	3	-	-	-	-
HL Indirect	STR Xl, (HL)	216	1	2	-	-	-	-

Store Yh

Operation: Memory ← Yh
Structure: 2 – 2 – Store Destination

Stores the value in Register Yh to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR Yh, $hhll	224	3	4	-	-	-	-
Zero Page	STR Yh, $zz	225	2	3	-	-	-	-
HL Indirect	STR Yh, (HL)	226	1	2	-	-	-	-

Store Yl

Operation: Memory ← Yl
Structure: 2 – 3 – Store Destination

Stores the value in Register Yl to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR Yl, $hhll	234	3	4	-	-	-	-
Zero Page	STR Yl, $zz	235	2	3	-	-	-	-
HL Indirect	STR Yl, (HL)	236	1	2	-	-	-	-

Store H

Operation: Memory ← H
Structure: 2 – 4 – Store Destination

Stores the value in Register H to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR H, $hhll	244	3	4	-	-	-	-
Zero Page	STR H, $zz	245	2	3	-	-	-	-

Store L

Operation: Memory ← L
Structure: 2 – 5 – Store Destination

Stores the value in Register L to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR L, $hhll	254	3	4	-	-	-	-
Zero Page	STR L, $zz	255	2	3	-	-	-	-

Store Acc

Operation: Memory ← Acc
Structure: 2 – 6 – Store Destination

Stores the value in the Accumulator to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR Acc, $hhll	264	3	4	-	-	-	-
Zero Page	STR Acc, $zz	265	2	3	-	-	-	-
HL Indirect	STR Acc, (HL)	266	1	2	-	-	-	-

Store SP

Operation: Memory ← SP
Structure: 2 – 7 – Store Destination

Stores the value in the Stack Pointer to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Absolute	STR SP, $hhll	274	3	4	-	-	-	-
Zero Page	STR SP, $zz	275	2	3	-	-	-	-
HL Indirect	STR SP, (HL)	276	1	2	-	-	-	-

Move Instructions

Move Operations

Overview

Common Structure: 3 – Source – Destination
This logic block contains solely move operations. Both Source and Destination designations are the same.

Registers:

[0] Xh Register
[1] Xl Register
[2] Yh Register
[3] Yl Register
[4] H Register
[5] L Register
[6] Accumulator
[7] Stack Pointer

Move Xh

Operation: Destination ← Xh
Structure: 3 – Destination – 0

Moves the value in Register Xh to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV Xh, Xl	310	1	2	-	-	-	-
Register	MOV Xh, Yh	320	1	2	-	-	-	-
Register	MOV Xh, Yl	330	1	2	-	-	-	-
Register	MOV Xh, H	340	1	2	-	-	-	-
Register	MOV Xh, L	350	1	2	-	-	-	-
Register	MOV Xh, Acc	360	1	2	-	-	-	-
Register	MOV Xh, SP	370	1	2	-	-	-	-

Move Xl

Operation: Destination ← Xl
Structure: 3 – Destination – 1

Moves the value in Register Xl to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV Xl, Xh	301	1	2	-	-	-	-
Register	MOV Xl, Yh	321	1	2	-	-	-	-
Register	MOV Xl, Yl	331	1	2	-	-	-	-
Register	MOV Xl, H	341	1	2	-	-	-	-
Register	MOV Xl, L	351	1	2	-	-	-	-
Register	MOV Xl, Acc	361	1	2	-	-	-	-
Register	MOV Xl, SP	371	1	2	-	-	-	-

Move Yh

Operation: Destination ← Yh
Structure: 3 – Destination – 2

Moves the value in Register Yh to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV Yh, Xh	302	1	2	-	-	-	-
Register	MOV Yh, Xl	312	1	2	-	-	-	-
Register	MOV Yh, Yl	332	1	2	-	-	-	-
Register	MOV Yh, H	342	1	2	-	-	-	-
Register	MOV Yh, L	352	1	2	-	-	-	-
Register	MOV Yh, Acc	362	1	2	-	-	-	-
Register	MOV Yh, SP	372	1	2	-	-	-	-

Move Yl

Operation: Destination ← Yl
Structure: 3 – Destination – 3

Moves the value in Register Yl to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV Yl, Xh	303	1	2	-	-	-	-
Register	MOV Yl, Xl	313	1	2	-	-	-	-
Register	MOV Yl, Yh	323	1	2	-	-	-	-
Register	MOV Yl, H	343	1	2	-	-	-	-
Register	MOV Yl, L	353	1	2	-	-	-	-
Register	MOV Yl, Acc	363	1	2	-	-	-	-
Register	MOV Yl, SP	373	1	2	-	-	-	-

Move H

Operation: Destination ← H
Structure: 3 – Destination – 4

Moves the value in Register H to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV H, Xh	304	1	2	-	-	-	-
Register	MOV H, Xl	314	1	2	-	-	-	-
Register	MOV H, Yh	324	1	2	-	-	-	-
Register	MOV H, Yl	334	1	2	-	-	-	-
Register	MOV H, L	354	1	2	-	-	-	-
Register	MOV H, Acc	364	1	2	-	-	-	-
Register	MOV H, SP	374	1	2	-	-	-	-

Move L

Operation: Destination ← L
Structure: 3 – Destination – 5

Moves the value in Register L to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV L, Xh	305	1	2	-	-	-	-
Register	MOV L, Xl	315	1	2	-	-	-	-
Register	MOV L, Yh	325	1	2	-	-	-	-
Register	MOV L, Yl	335	1	2	-	-	-	-
Register	MOV L, H	345	1	2	-	-	-	-
Register	MOV L, Acc	365	1	2	-	-	-	-
Register	MOV L, SP	375	1	2	-	-	-	-

Move Accumulator

Operation: Destination ← Acc
Structure: 3 – Destination – 6

Moves the value in the Accumulator to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV Acc, Xh	306	1	2	-	-	-	-
Register	MOV Acc, Xl	316	1	2	-	-	-	-
Register	MOV Acc, Yh	326	1	2	-	-	-	-
Register	MOV Acc, Yl	336	1	2	-	-	-	-
Register	MOV Acc, H	346	1	2	-	-	-	-
Register	MOV Acc, L	356	1	2	-	-	-	-
Register	MOV Acc, SP	376	1	2	-	-	-	-

Move Stack Pointer

Operation: Destination ← SP
Structure: 3 – Destination – 7

Moves the value in the Accumulator to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Register	MOV SP, Xh	307	1	2	-	-	-	-
Register	MOV SP, Xl	317	1	2	-	-	-	-
Register	MOV SP, Yh	327	1	2	-	-	-	-
Register	MOV SP, Yl	337	1	2	-	-	-	-
Register	MOV SP, H	347	1	2	-	-	-	-
Register	MOV SP, L	357	1	2	-	-	-	-
Register	MOV SP, Acc	367	1	2	-	-	-	-

Extended Instruction Set

Indirect Arithmetic and Logic,
Load, Store, Move, and
Double Register Stack Operations

Overview

Common Structure: 0 – Row – Operations
This logic block, similarly to Logic Block 0, contains miscellaneous operations which utilize the Xhl and Yhl Combined Registers as memory pointers.

Operations:

- [0] Xhl Indirect Arithmetic and Logic
- [1] Yhl Indirect Arithmetic and Logic
- [2] Xhl Indirect Load
- [3] Yhl Indirect Load
- [4] Xhl Indirect Store
- [5] Yhl Indirect Store
- [6] 16 Bit Move Instructions
- [7] 16 Bit Push/Pop

Arithmetic and Logic Operation Rows:

- [0] Add
- [1] Add with Carry
- [2] Subtract
- [3] Subtract with Carry
- [4] Compare
- [5] AND
- [6] OR
- [7] XOR

Load/Store Rows:

- [0] Xh Register
- [1] Xl Register
- [2] Yh Register
- [3] Yl Register
- [4] H Register
- [5] L Register
- [6] Accumulator
- [7] Stack Pointer

16 Bit Move Instruction Rows:

[0] Move Xhl ← Yhl
[1] Move Xhl ← HL
[2] Move Yhl ← Xhl
[3] Move Yhl ← HL
[4] Move HL ← Xhl
[5] Move HL ← Yhl

16 Bit Push/Pop Rows:

[0] Push Xhl
[1] Pop Xhl
[2] Push Yhl
[3] Pop Yhl
[4] Push HL
[5] Pop HL
[6] Push PC
[7] Pop PC

Ext. Add

Operation: Acc, C, N, Z ← Acc + Operand
Structure: 0 – Operand – 0/1

Adds the value in the Accumulator to the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	ADD (Xhl)	000	1	2	✓	✓	✓	-
Yhl Indirect	ADD (Yhl)	001	1	2	✓	✓	✓	-

Ext. Add with Carry

Operation: Acc, C, N, Z ← Acc + Operand + C
Structure: 0 – Operand – 0/1

Adds the value in the Accumulator to the Operand with Carry In.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	ADC (Xhl)	010	1	2	✓	✓	✓	-
Yhl Indirect	ADC (Yhl)	011	1	2	✓	✓	✓	-

Ext. Subtract

Operation: Acc, C, N, Z ← Acc - Operand
Structure: 0 – Operand – 0/1

Subtracts the value in the Accumulator to the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	SUB (Xhl)	020	1	2	✓	✓	✓	-
Yhl Indirect	SUB (Yhl)	021	1	2	✓	✓	✓	-

Ext. Subtract with Borrow

Operation: Acc, C, N, Z ← Acc - Operand - C
Structure: 0 – Operand – 0/1

Subtracts the value in the Accumulator to the Operand with Carry In.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	SBB (Xhl)	030	1	2	✓	✓	✓	-
Yhl Indirect	SBB (Yhl)	031	1	2	✓	✓	✓	-

Ext. Compare

Operation: C, N, Z ← Acc - Operand
Structure: 0 – Operand – 0/1

Subtracts the value in the Accumulator to the Operand, but does not save the difference.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	CMP (Xhl)	040	1	2	✓	✓	✓	-
Yhl Indirect	CMP (Yhl)	041	1	2	✓	✓	✓	-

Ext. AND Operation

Operation: Acc, C, N, Z ← Acc AND Operand
Structure: 0 – Operand – 0/1

Performs the AND operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	AND (Xhl)	050	1	2	✓	✓	✓	-
Yhl Indirect	AND (Yhl)	051	1	2	✓	✓	✓	-

Ext. OR Operation

Operation: Acc, C, N, Z ← Acc OR Operand
Structure: 0 – Operand – 0/1

Performs the OR operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	OR (Xhl)	060	1	2	✓	✓	✓	-
Yhl Indirect	OR (Yhl)	061	1	2	✓	✓	✓	-

Ext. XOR Operation

Operation: Acc, C, N, Z ← (Acc OR Operand) AND NOT (Acc AND Operand)
Structure: 0 – Operand – 0/1

Performs the XOR operation on the Accumulator and the Operand.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	XOR (Xhl)	070	1	2	✓	✓	✓	-
Yhl Indirect	XOR (Yhl)	071	1	2	✓	✓	✓	-

Ext. Load Xh

Operation: Xh ← Memory
Structure: 2 – Load Source – 3

Loads a value from Memory into Register Xh.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Yhl Indirect	LD Xh, (Yhl)	003	1	2	-	-	-	-

Ext. Load Xl

Operation: Xl ← Memory
Structure: 2 – Load Source – 3

Loads a value from Memory into Register Xl.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Yhl Indirect	LD Xh, (Yhl)	013	1	2	-	-	-	-

Ext. Load Yh

Operation: Yh ← Memory
Structure: 2 – Load Source – 2

Loads a value from Memory into Register Yh.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	LD Yh, (Xhl)	022	1	2	-	-	-	-

Ext. Load Yl

Operation: Yl ← Memory
Structure: 2 – Load Source – 2

Loads a value from Memory into Register Yl.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	LD Yl, (Xhl)	032	1	2	-	-	-	-

Ext. Load H

Operation: H ← Memory
Structure: 2 – Load Source – 2/3

Loads a value from Memory into Register H.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	LD H, (Xhl)	042	1	2	-	-	-	-
Yhl Indirect	LD H, (Yhl)	043	1	2	-	-	-	-

Ext. Load L

Operation: L ← Memory
Structure: 2 – Load Source – 2/3

Loads a value from Memory into Register L.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	LD L, (Xhl)	052	1	2	-	-	-	-
Yhl Indirect	LD L, (Yhl)	053	1	2	-	-	-	-

Ext. Load Accumulator

Operation: Acc ← Memory
Structure: 2 – Load Source – 2/3

Loads a value from Memory into the Accumulator.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	LD Acc, (Xhl)	062	1	2	-	-	-	-
Yhl Indirect	LD Acc, (Yhl)	063	1	2	-	-	-	-

Ext. Store Xh

Operation: Memory ← Xh
Structure: 2 – Store Destination – 5

Stores the value in Register Xh to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Yhl Indirect	STR Xh, (Yhl)	005	1	2	-	-	-	-

Ext. Store Xl

Operation: Memory ← Xl
Structure: 2 – Store Destination – 5

Stores the value in Register Xl to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Yhl Indirect	STR Xl, (Yhl)	015	1	2	-	-	-	-

Ext. Store Yh

Operation: Memory ← Yh
Structure: 2 – Store Destination – 4

Stores the value in Register Yh to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	STR Yh, (Xhl)	024	1	2	-	-	-	-

Ext. Store Yl

Operation: Memory ← Yl
Structure: 2 – Store Destination – 4

Stores the value in Register Yl to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	STR Yl, (Xhl)	034	1	2	-	-	-	-

Ext. Store H

Operation: Memory ← H
Structure: 2 – Store Destination – 4/5

Stores the value in Register H to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	STR H, (Xhl)	044	1	2	-	-	-	-
Yhl Indirect	STR H, (Yhl)	045	1	2	-	-	-	-

Ext. Store L

Operation: Memory ← L
Structure: 2 – Store Destination – 4/5

Stores the value in Register L to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	STR L, (Xhl)	054	1	2	-	-	-	-
Yhl Indirect	STR L, (Yhl)	055	1	2	-	-	-	-

Ext. Store Acc

Operation: Memory ← Acc
Structure: 2 – Store Destination – 4/5

Stores the value in the Accumulator to Memory.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Xhl Indirect	STR Acc, (Xhl)	064	1	2	-	-	-	-
Yhl Indirect	STR Acc, (Yhl)	065	1	2	-	-	-	-

Ext. Move Xhl

Operation: Destination ← Xhl
Structure: N/A

Moves the value in Double Register Xhl to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	MOV Xhl, Yhl	026	1	3	-	-	-	-
Double Register	MOV Xhl, HL	046	1	3	-	-	-	-

Ext. Move Yhl

Operation: Destination ← Yhl
Structure: N/A

Moves the value in Double Register Yhl to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	MOV Yhl, Xhl	006	1	3	-	-	-	-
Double Register	MOV Yhl, HL	056	1	3	-	-	-	-

Ext. Move HL

Operation: Destination ← HL
Structure: N/A

Moves the value in Double Register HL to the Destination.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	MOV HL, Xhl	016	1	3	-	-	-	-
Double Register	MOV HL, Yhl	036	1	3	-	-	-	-

Ext. Push

Operation: Stack ← Double Register, SP ← SP + 2
Structure: N/A

Pushes the Double Register specified in the instruction onto the Stack.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	PSH Xhl	007	1	6	-	-	-	-
Double Register	PSH Yhl	027	1	6	-	-	-	-
Double Register	PSH HL	047	1	6	-	-	-	-
Double Register	PSH PC	067	1	6	-	-	-	-

Ext. Pop

Operation: SP ← SP – 2, Stack ← Double Register
Structure: N/A

Pops the two topmost values from the Stack into the Double Register specified in the instruction.

Addr. Mode	Assembly	Opcode	Bytes	Cycles	C	N	Z	I
Double Register	POP Xhl	017	1	7	-	-	-	-
Double Register	POP Yhl	037	1	7	-	-	-	-
Double Register	POP HL	057	1	7	-	-	-	-
Double Register	POP PC	077	1	7	-	-	-	-