eZ80 CPU

User Manual

UM007701-1000

ZiLOG WORLDWIDE HEADQUARTERS • 910 E. HAMILTON AVENUE • CAMPBELL, CA 95008 USA
PHONE: 1.408.558.8500 • FAX: 1.408.558.8300 • WWW.ZILOG.COM

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
910 E. Hamilton Avenue
Campbell, CA 95008 USA
Telephone: 1.408.558.8500
Fax: 1.408.558.8300
www.ZiLOG.com

Document Disclaimer

Table of Contents

List of Tables


ZiLOG

Introduction

The eZ80 is the fastest 8-bit CPU available today, executing code four times faster than a standard Z80 operating at the same clock speed. The increased processing efficiency can be used to improve available bandwidth or to decrease power consumption. Considering both the increased clock speed and processor efficiency, the eZ80's processing power rivals the performance of 16-bit microprocessors.

The eZ80 is also the first 8-bit microprocessor to support 16MB linear addressing. Each software module, or each task under a real-time executive or operating system, can operate in Z80-compatible (64KB) mode or full 24-bit (16MB) address mode.

Architectural Overview

The eZ80 is ZiLOG's next-generation Z80 processor core. It is the basis of a new family of integrated microprocessors and includes the following features:

- Upward-code-compatible from Z80 and Z180
- Several address-generation modes, including 24-bit linear addressing
- 24-bit registers and ALU
- Single-cycle fetch

Processor Description

The eZ80 is an 8-bit microprocessor that performs certain 16- or 24-bit operations. The processor includes two accumulators. Register A is the accumulator for 8-bit operations and the multi-byte register HL is the accumulator for 16- and 24-bit operations.

Processor Registers

In addition to register A, there are six more 8-bit registers (B, C, D, E, H, and L) that are part of the BC, DE, and HL multi-byte registers. Flag register F completes the basic register bank. There are two basic register banks, as described above. The Main register set and the Alternate register set. High-speed exchange between these banks can be used by a program internally or one bank can be allocated to the mainline program and the other to interrupt service routines.

- Program Counter (PC)—The Program Counter holds the address of the current instruction being fetched.
- Stack Pointer (SP)—The stack pointer holds the address for the current top of a stack located anywhere in memory. The external stack is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack or popped off of the stack using the **PUSH** and **POP** instructions. The **CALL** and **RET** instructions also use the stack for program address storage. The eZ80 has a 24-bit Stack Pointer (SPL) and a 16-bit Stack Pointer (SPS).
- Two Index Registers (IX and IY)—These registers allow base and displacement addressing in memory. IX and IY are not included in the register banks on the eZ80. They are independent of the register banks.
- Interrupt Vector Register (I)—This register holds the upper eight bits of the interrupt vector table address for Mode 2 vectored interrupts.

eZ80 CPU Registers

In non-ADL mode, the BC, DE and HL register pairs and IX and IY registers normally function as 16-bit registers for multi-byte operations and indirect addressing. The high-order BCU, DEU, HLU, IXU, and IYU bits are usually not used in non-ADL mode. The active Stack Pointer is SPS and the Program Counter (PC) is 16 bits long. Address (23-16) comes from the MBASE register, which can only be set to 1 while in ADL mode.

	A	F
BCU	B	C
DEU	D	E
HLU	H	L

Main Register Set

	A’	F’
BCU’	B’	C’
DEU’	D’	E’
HLU’	H’	L’

Alternate Register Set

I	R	MBASE

IXU	IX
IYU	IY
	SPS
SPL	
	PC

IEF1	IEF2
ADL	MADL

In ADL mode, the BC, DE, and HL multi-byte registers are 24 bits long, as are the IX and IY registers. MBASE is not used for address generation in ADL mode, but can be written only in ADL mode. The Program Counter is 24 bits long, as is the Stack Pointer (SPL). IEF1, IEF2, ADL, and MADL are single bit flags.

	A	F
BCU	B	C
DEU	D	E
HLU	H	L

Main Register Set

	A’	F’
BCU’	B’	C’
DEU’	D’	E’
HLU’	H’	L’

Alternate Register Set

I	R	MBASE

IEF1	IEF2
ADL	MADL

IX
IY
SPS
SPL
PC

Operating Modes

The eZ80 has 16-and 24-bit addressing modes that are controlled by the Address and Data Long (ADL) bit.

When ADL is cleared to 0:

> The PC, SP, BC, DE, HL, IX, and IY registers are effectively 16 bits wide, as in the Z80 and Z80180.

When ADL is set to 1:

> The PC, SP, BC, DE, HL, IX, and IY registers are 24 bits wide.

The multiple operating modes of the processor allows Z80 code to be run without change in native Z80 or virtual Z80 with ADL cleared to zero. If ADL is set to one, the application can take advantage of the eZ80's 16MB linear addressing space and enhanced instruction set.

Native Z80 Mode

ADL and MBASE reset to 0. In this Native Z80 state, the programming model includes 16-bit registers and addresses and a 64KB memory space at the beginning of the eZ80's potential 16MB memory space. The upper eight bits of address (23-16) are held at zero, which is the value of MBASE. This is the mode the eZ80 starts in after reset.

Virtual Z80 Mode

If ADL is cleared to zero, but MBASE contains a non-zero value, the programming model still includes 16-bit registers and a 64KB memory space, but this space is relocated in the 16MB memory space by MBASE. The upper eight bits of address (23-16) are the value of the MBASE register. In this Virtual Z80 mode, several tasks can have their own Z80 partition. The MBASE register can only be changed while in ADL mode.

ADL Mode

If ADL is set to 1, MBASE has no effect on memory addressing. In this mode, the PC, BC, DE, HL, IX and IY registers are expanded from 16 to 24 bits and a 24-bit Stack Pointer Long (SPL) register replaces the 16-bit Stack Pointer Short (SPS) register that is used in the other modes. When the processor fetches an instruction that includes a 16-bit address or immediate data in the other modes, it automatically fetches a 24-bit address or data.

Mode Switching

The eZ80 only switches between ADL mode and the other modes as part of a specially-prefixed **CALL**, **JP**, **RET**, or **RST** instruction, or an interrupt or trap operation. The MBASE register can only be changed while in ADL mode.

Interrupts

Interrupts allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start a peripheral service routine. Once the service routine is completed, the CPU returns to the operation in which it was interrupted.

Interrupt Enable/Disable

The eZ80 has three interrupt inputs, two software maskable interrupts and a non-maskable interrupt. The non-maskable interrupt (NMI) cannot be disabled by the programmer, but is accepted when the peripheral device requests it. You can enable or disable the maskable interrupts (INT and INTV).

In the eZ80 CPU, there are two interrupt enable flags (called IEF1 and IEF2) that you can set or reset using the Enable Interrupt (**EI**) and Disable Interrupt (**DI**) instructions. When IEF1 is reset, a maskable interrupt cannot be accepted by the CPU.

The state of IEF1 is used to inhibit interrupts, while IEF2 is used as a temporary storage location for IEF1. At reset, the CPU forces the state of both IEF1 and IEF2 to the reset state to disable interrupts. They can be enabled using the **EI** instruction. No pending interrupt is accepted until the instruction that follows the **EI** instruction. The single instruction delay occurs because **EI** is followed by a return instruction and interrupts must not be allowed until the return is complete. The **EI** instruction sets both IEF1 and IEF2 to the enable state. The **DI** instruction resets both IEF1 and IEF2 to the disabled state. When a maskable interrupt is accepted by the CPU, both IEF1 and IEF2 are reset to the disabled state, thus inhibiting further interrupts until you issue a new **EI** instruction. For all of the previous cases, IEF1 and IEF2 are always equal.

The purpose of IEF2 is to save the status of IEF1 when a non-maskable interrupt occurs. When a non-maskable interrupt is accepted, IEF1 is reset to prevent further interrupts until you reenable them. The **LD A,I** or **LD A,R** instructions copy the state of IEF2 into the Parity Flag where it can be tested or stored. The status of IEF1 can also be restored by executing the Return From Non Maskable (**RETN**) instruction. The contents of IEF2 is copied back into IEF1.

CPU Response

Non-Maskable Interrupt

The CPU always accepts a non-maskable interrupt (NMI). When this occurs, the CPU ignores the next instruction that it fetches and instead does a restart to location 0066H.

Maskable Interrupts

The eZ80 can be programmed to respond to a maskable interrupt in Mode 0, 1, or 2.

Mode 0

With this mode, the interrupting device places the next instruction onto the data bus during the interrupt acknowledge cycle. This instruction is typically a single byte restart (**RST**) instruction. After external reset, the CPU enters Mode 0.

Mode 1

When you select this mode, the CPU responds to an interrupt by executing a restart to location 0038H.

Mode 2

This is the most powerful interrupt response mode. With this mode, you can maintain a table of 16-bit starting addresses for every interrupt service routine. This table may be located anywhere within the first 64K of memory. When an interrupt is accepted, the interrupting device places the lower eight bits of the interrupt vector on the data bus during the interrupt acknowledge cycle (Bit 0 is assumed to be zero). The actual vector address consists of Bits 23-16 as zero, Bits 15-8 are the contents of the I register, and Bits 7-0 are supplied vector byte.

The 16-bit word at the above vector address is fetched and its value is the beginning of the interrupt service routine with Bits 23-16 provided by the eZ80 processor as zeros.

In ADL mode, the interrupt table must always be in the first 64KB, as must the start of interrupt service routines entered through the interrupt table.

Vectored Interrupts

Vectored interrupts operate the same as Mode 2 interrupts no matter which interrupt mode is selected. In the case of the vectored interrupt response, the CPU fetches the low-order interrupt vector address not from the data bus, but from the ivedct bus. The vectored interrupt source is used exclusively for on-chip peripherals.

Illegal Instruction Traps

The eZ80 instruction set does not cover all possible sequences of binary values. Sequences for which no operation is defined, are illegal instructions. When an eZ80 processor fetches one of these sequences, it performs a Trap sequence. The byte of the multi-byte instruction that caused the trap is indicated by the Trap register. The Trap register is not part of the eZ80 CPU, so you must refer to the *eZ80 Product Specification* for its Trap register configuration. An instruction trap resets the Program Counter (PC) to zero.

Interrupts and Traps

Applications that only operate in Native Z80 mode or ADL mode are relatively simple with respect to interrupts and traps. In these modes, memory always starts at the beginning of the eZ80's potential 16MB memory space and the interrupt and trap locations are never mapped. This means applications that switch between modes or operate in Virtual Z80 mode, can simplify interrupts and trap-handling by executing a **STMIX** instruction to set the mixed ADL bit. If the mixed ADL bit is set to 1, interrupts and instruction traps stack the ADL state, as well as the PC, and enter ADL mode in the first 64KB of the eZ80's potential 16MB memory space.

I/O Space

A separate I/O space may include both on- and off-chip peripheral devices. The eZ80 features a 64K I/O space with 16-bit addresses.

Memory

The eZ80 provides three address generation modes—Native Z80, Virtual Z80 and ADL mode.

Native Z80 Mode

The total memory address space is the first 64KB of the overall eZ80 memory space. The Memory Base (MBASE) register is zero.

Virtual Z80 Mode

The memory address space can be any 64KB in the overall 16MB eZ80 memory space under the control of the MBASE register.

ADL Mode

The eZ80 operates in a 16MB linear address space. In this mode, the 16-bit PC, BC, DE, HL, IX, and IY registers expand to 24 bits. The processor automatically fetches an additional byte of address or immediate data in those instructions that contain a 16-bit address or data in other modes. Prefix-override bytes allow any instruction to operate in ADL mode in one of the first two modes or to use MBASE addressing in ADL mode.

Addressing Modes

Memory addresses can be formed in several ways. The eZ80 addressing modes include:

- Immediate—In this mode of addressing, the byte(s) following the Op Code in memory contains the actual operand. When the immediate operand is larger than eight bits, the low-order byte is next after the Op Code byte followed by the high-ordered bytes.
- Modified Page Zero—The eZ80 has a special single byte **CALL** instruction to any of the eight locations in page zero (the first 256 bytes) of memory. These restart (**RST**) instructions set the PC to an effective address in page zero.

- Relative Address—Relative addressing uses one byte of data following the Op Code to specify a displacement from the existing program to which a program jump can occur. The displacement is a signed two's complement number that is added to the address of the Op Code following the instruction. The signed displacement can range from +127 to -128. This specifies a range of addresses -126 to +129 from the start of the instruction to which program control can be transferred.
- Direct Address—Direct-addressing instructions include a 16-bit logical or 24-bit linear address, depending on a prefix byte or the ADL mode. This can be an address to which the program can jump to or the address where the operand is located.
- Indexed Addressing—In this mode of addressing, a byte of data following the Op Code contains a displacement to be added to one of the IX or IY index registers. The displacement is in two's complement form in the range of +127 to -128. Depending on a prefix byte or the ADL mode, the index register supplies a 24-bit linear address or a 16-bit logical address that is subject to MBASE.
- Register Indirect Addressing—The address is taken from one of the multi-byte BC, DE or HL registers. Depending on a prefix byte or the ADL mode, the register supplies a 24-bit linear address or a 16-bit logical address that is subject to MBASE.
- Register Addressing—Many of the eZ80 instructions contain information that specifies the CPU register that is to be used for an operation.

ADL and Mixed ADL

Interrupts, Traps, and RST Instructions

All of these operations are affected by a global state called Mixed ADL, which should be set appropriately for each application. Mixed ADL should be **0** for applications in which all code runs in the same ADL state, but should be set to 1 for applications that include some code that runs in ADL mode and some that runs in other modes. If Mixed ADL is set to 1, and an interrupt, trap, or restart occurs, the eZ80 stacks a byte containing the ADL mode of the interrupted, trapped, or called process on SPL before setting ADL mode for the service routine.

Changing the ADL Mode: CALL, RST, JP, and RET

There is no separate instruction to simply change ADL because after such an instruction the Program Counter would undergo an unmanageable change in interpretation. ADL can be changed only by prefixing a **CALL**, **JP NNNN**, **RST**, **RET** or **JP (RR)** instruction with an ADL modifying prefix. The following tables describe these instructions for various cases of prefix bytes and the ADL mode.

Table 1. CALL Instruction

ADL	Prefix	Operation
0	None	Stack 2-byte logical return address using SPS mapped by MBASE. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	None	Stack the 3-byte return address using SPL. Keep ADL set to 1. Load a 3-byte address from the instruction into PC.
0	.IS	Stack 2-byte logical return address using SPS mapped by MBASE. Stack a 00 byte using SPL. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	.IS	Stack the 2 LS bytes of the return address using SPS mapped by MBASE. Stack the MS byte of the return address using SPL. Stack a 01 byte using SPL. Clear ADL to 0. Load a 2-byte logical address from the instruction into PC.
0	.IL	Stack the 2-byte logical return address using SPL. Stack a 00 byte using SPL. Set ADL to 1. Load a 3-byte address from the instruction into PC.
1	.IL	Stack the 3-byte return address using SPL. Stack a 01 byte using SPL. Keep ADL set to 1. Load a 3-byte address from the instruction into PC.

Table 2. RST nn Instruction

ADL	Prefix	Operation
0	None	Stack 2-byte logical return address using SPS mapped by MBASE. Keep ADL cleared to 0. Load the 16-bit logical address 00nn into PC.
1	None	Stack the 3-byte return address using SPL. Keep ADL set to 1. Load the 24-bit address 0000nn into PC.
0	.IS	Stack 2-byte logical return address using SPS mapped by MBASE. Stack a 00 byte using SPL. Keep ADL cleared to 0. Load the 16-bit logical address 00nn into PC.
1	.IS	Stack the 2 LS bytes of the return address using SPS mapped by MBASE. Stack the MS byte of the return address using SPL. Stack a 01 byte using SPL. Clear ADL to 0. Load the 16-bit logical address 00nn into PC..
0	.IL	Stack the 2-byte logical return address using SPL. Stack a 00 byte using SPL. Set ADL to 1. Load the 24-bit address 0000nn into PC.
1	.IL	Stack the 3-byte return address using SPL. Stack a 01 byte using SPL. Keep ADL set to 1. Load the 24-bit address 0000nn into PC.

Table 3. JP nnnn Instruction

ADL	Prefix	Operation
0	None	Load a 2-byte logical address from the instruction into PC. Keep ADL cleared to 0.
1	None	Load a 3-byte address from the instruction into PC. Keep ADL set to 1.
x	.SIS	Clear ADL to 0. Load a 2-byte logical address from the instruction into PC.
x	.LIL	Set ADL to 1. Load a 3-byte address from the instruction into PC.

Table 4. RET, RETI, RETN Instruction

ADL	Prefix	Operation
0	None	Pop a 2-byte logical address from SPS mapped by MBASE into PC. Keep ADL cleared to 0.
1	None	Pop a 3-byte logical address from SPL into PC. Keep ADL set to 1.
0	.S or .L	Pop a byte from SPL. Load its units bit into ADL. If ADL is still cleared to 0, Pop 2-byte logical address from SPS mapped by MBASE into PC. If ADL is now set to 1, Pop a byte from SPL into PC23-16. Then pop two bytes from SPS mapped by MBASE into PC15-0.
1	.S or .L	Pop a byte from SPL/ Load its units bit into ADL/ If ADL is now cleared to 0, pop a 2-byte logical address from SPL into PC. If ADL is still set to 1, pop a 3-byte address from SPL into PC.

Table 5. JP (rr) Instruction

ADL	Prefix	Operation
0	None	Load a 16-bit logical address from the register into PC. Keep ADL cleared to 0.
1	None	Load a 24-bit address from the register into PC. Keep ADL set to 1.
x	.IS	Clear ADL to 0. Load a 16-bit logical address from the register into PC.
x	.IL	Set ADL to 1. Load a 24-bit address from the register into PC.

Mixed-ADL Applications

Applications that include legacy routines/functions/tasks/modules that run in non-ADL mode and new routines/functions/tasks/modules that run in ADL mode, must follow certain rules to ensure proper operation:

- Include a **STMIX** instruction in device initialization to ensure that interrupt service routines begin in a consistent mode (ADL mode).
- End all interrupt service routines with a prefixed **RET** or **RETI** instruction, which pops the interrupted code's ADL state from the SPL stack.
- **CALL** or **JP** to each routine/function/task/module in the mode in which it was assembled or compiled.
- Any routine that may be called from either mode must be called with a prefix to save the caller's ADL mode on the SPL stack.
- Any routine that may be called from either mode must return with a prefixed **RET** instruction to restore the caller's ADL state from the SPL stack.
- If a calling code operating in one mode must pass stack-based operands/ arguments to a routine compiled or assembled for a different mode, it must use prefixed instructions to set up the operands/arguments. For **PUSH**, .S and .L prefixes control whether SPS or SPL is used and whether the operands/ arguments are stored as 2- or 3-byte values.

Note: In mixed-ADL applications, some of the rules above may represent exceptions to the eZ80's design goal that legacy code not have to be modified to be run on the eZ80. Assuming that legacy routines are not selectively converted to ADL mode and that legacy routines don't call newly written routines, the only rule that would lead to such modification would be the fifth rule above. If each legacy routine ends with a single **RET** at its end, this conversion is easy. Internal and conditional **RET**s require more careful review -- a program to highlight **RET**s may be helpful.

Prefix Bytes: Exceptions to ADL Mode

In the ZiLOG ZMASM / ZDS assembler, code is assembled for a given state of the ADL mode bit by preceding it with a pseudo-op:

```
.assume adl=1 ; or adl=0
```

You are, of course, responsible for ensuring that this source file setting matches the state of the hardware ADL mode bit when the code is executed.

The ADL mode and assumed setting govern several different aspects of instruction and eZ80 operation, as described in preceding sections. Two different kinds of exceptions to normal operation can be selected for a particular instruction by adding a suffix to the instruction’s Op Code.

Suffixes .IS and .IL control whether a memory address or multi-byte immediate data in the instruction should be two or three bytes long. .S and .L control whether the overall operation of the instruction should involve 16 or 24 bits.

A few instructions that involve both a multi-byte register and a direct memory address or immediate data are affected by both exceptions. The proper suffix for exceptions on these instructions is .SIS, .SIL, .LIS, or .LIL.

Special care must be taken when using the .SIL suffix. This case assembles a long direct memory address or immediate data in the instruction stream and the CPU reads in the 24-bit value. But as the .S is active, the internal registers are treated as 16-bit registers and the upper bits (23-16) that were read from the instruction are discarded.

Note: The assembler generates the .SIL prefix byte when in non-ADL mode and the .IL suffix is used. For example, in non-ADL mode, the instruction **JP.IL** 123456h results in the PC(15-0) loaded with 3456H, PC(23-16) are zero, and the jump is to 3456H offset by MBASE.

For the sake of those instructions, the prefix bytes always express both kinds of exceptions. The prefix bytes replace several Z80 and Z80180 instructions that have no function. If an eZ80 assembler encounters one of these replaced instructions, it issues a warning message and assemble it as a standard NOP (00H). The eZ80 prefix bytes are shown in the following table.

Op Code Prefix	Z80 Instruction	eZ80 Suffix
40H	LD B,B	.SIS
49H	LD C,C	.LIS
52H	LD D,D	.SIL
5BH	LD E,E	.LIL

As for the traditional Z80 prefix bytes, the eZ80 does not allow an interrupt to occur in the time between fetching one of these prefix bytes and fetching the following instruction. These prefix bytes must precede traditional Z80 prefix bytes.

Reset Conditions

The effects of reset on ADL, Mixed ADL, MBASE, PC, SP, I, IEF1, IEF2, R, and F are cleared to zero. A, B, C, D, E, H, L, IX, and IY are not changed by reset.

Instruction Set

Assembly Language

The eZ80 assembly language is designed to minimize the number of different Op Codes corresponding to the set of basic machine operations, as well as to provide a consistent description of instruction operands. The nomenclature has been defined with special emphasis on mnemonic values and readability.

The movement of data is indicated by a single Op Code, regardless of whether the movement is between different registers or between registers and memory locations.

For example, the first operand of an **LD** instruction is the destination of the operation and the second operand is the source of the operation. For example, **LD A, B** indicates that the contents of the second operand, register B, are to be transferred to the first operand, register A. Similarly, **LD C**, 3FH indicates that the constant 3FH is to be loaded into Register C. In addition, enclosing an operand in parentheses indicates a memory location addressed by the contents of the parentheses. For example, **LD HL**, (1200) indicates the contents of memory locations 1200 and 1201 that are to be loaded into the 16-bit register pair HL. Similarly, **LD (IX+6),C** indicates that the contents of register C are to be stored in the memory location addressed by the current value in the 16-bit IX register plus 6.

eZ80 Status Indicators

The Flag register (F and F') contains the status of the eZ80. The bit position for each flag is as follows:.

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

where:

C = Carry Flag
N = Add/Subtract Flag
P/V = Parity/Overflow Flag
H = Half-Carry Flag
Z = Zero Flag
S = Sign Flag
X = Not used

Each of the two eZ80 Flag registers contain six bits of status information that are set or reset by CPU operations. Bits 3 and 5 are not used. Four of these bits are testable (C, P/V, Z and S) for use with conditional **JUMP**, **CALL** or **RETURN** instructions. Two flags are not testable (H, N) and are used for BCD arithmetic.

Carry Flag

The Carry bit is set or reset, depending on the operation that is performed. For **ADD** instructions that generate a carry and **SUBTRACT** instructions that generate a borrow, the Carry Flag is set to 1. The Carry Flag is reset by an **ADD** that does not generate a carry and a subtract that does not generate a borrow. This saved carry facilitates software routines for extended precision arithmetic. Also, the **DAA** instruction sets the Carry Flag to 1 if the conditions for making the decimal adjustment are met.

For the **RLA**, **RRA**, **RLS** and **RRS** instructions, the Carry bit is used as a link between the least-significant bit (LSB) and most-significant bit (MSB) for any register or memory location. During the **RLCA**, **RLC m** and **SLA m** instructions, the carry contains the last value shifted out of Bit 7 of any register or memory location. During the **RRCA**, **RRC m**, **SRA m**and **SRL m** instructions, the carry contains the last value shifted out of Bit 0 of any register or memory location. For the logical instructions **AND A s**, **OR A s**, and **XOR A s**, the carry is reset. The Carry Flag can also be set (SCF) and complemented (CCF).

Add/Subtract Flag

The Add/Subtract (N) Flag is used by the decimal adjust accumulator instructions (**DAA**) to distinguish between **ADD** and **SUBTRACT** instructions. For all **ADD** instructions, N is set to zero. For all **SUBTRACT** instructions, N is set to 1.

Parity/Overflow Flag

The Parity/Overflow (P/V) Flag is set or reset, depending on the operation that is performed. For arithmetic operations, this flag indicates an overflow condition when the result in the Accumulator is greater than the maximum possible number (+127) or is less than the minimum possible number (-128). This overflow condition can be determined by examining the sign bits of the operands.

For addition, operands with different signs never causes overflow. When adding operands with like signs and the result has a different sign, the overflow flag is set to 1, as shown in the following example.

+120	=	0111	1000	ADDEND
+105	=	0110	1001	AUGEND
+225		1110	0001	(-95) SUM

The two numbers added together result in a number that exceeds +127 and the two positive operands result in a negative number (-95), which is incorrect. Thus, the Overflow Flag is set to 1.

For subtraction, overflow can occur for operands of unlike signs. Operands of like signs never causes overflow, as shown in the following example.

	+127	0111	1111	MINUEND
(-)	-64	1100	0000	SUBTRAHEND
	+191	1011	1111	DIFFERENCE

The minuend sign has changed from positive to negative, giving an incorrect difference. Thus, overflow is set to 1. Another method for predicting an overflow is to observe the carry into and out of the sign bit. If there is a carry in and no carry out, then overflow has occurred. This flag is also used with logical operation and rotate instructions to indicate the parity of the result. The number of 1 bits in a byte are counted. If the total is odd, then odd parity (P=0) is flagged. If the total is even, then even parity (P=1) is flagged.

During search instructions (**CPI**, **CPIR**, **CPD**, **CPDR**) and block transfer instructions (**LDI**, **LDIR**, **LDD**, **LDDR**), the P/V Flag monitors the state of the byte count register (BC). When decrementing, the byte counter results in a zero value and the flag is reset to 0; otherwise the flag is logic 1.

During **LD A, I** and **LD A, R** instructions, the P/V Flag is set to 1 with the contents of the interrupt enable flip-flop (IEF2) for storage or testing. When inputting a byte from an I/O device, **IN r,(C)**, the flag is adjusted to indicate the parity of the data.

Half-Carry Flag

The Half-Carry (H) Flag is set or reset, depending on the carry and borrow status between Bits 3 and 4 of an 8-bit arithmetic operation. This flag is used by the decimal adjust accumulator instruction (**DAA**) to correct the result of a packed BCD addition or subtraction. The H flag is set to 1 or reset to 0, according to the following table.

H	ADD	SUBTRACT
1	There is a carry from Bit 3 to Bit 4	There is a borrow from Bit 4.
0	There is no carry from Bit 3 to Bit 4	There is no borrow from Bit 4.

Zero Flag

The Zero (Z) Flag is set to 1 or reset to 0 if the result generated by the execution of certain instructions is zero. For 8-bit arithmetic and logical operations, the Z Flag is set to 1 if the resulting byte in the Accumulator is 0. If the byte is not 0, the Z Flag is reset to 0.

For compare (search) instructions, the Z Flag is set to 1 if a comparison is found between the value in the Accumulator and the memory located pointed to by the contents of the register pair HL. When testing a bit in a register or memory location, the Z Flag contains the complemented state of the indicated bit (see the **BIT b, m** instruction).

When inputting or outputting a byte between a memory location and an I/O device (**INI**, **IND**, **OUTI** and **OUTD**), if the result of B-1 is zero, the Z Flag is set to 1. Otherwise, the Z Flag is reset. Also, for byte inputs from I/O devices using **IN r,(C)**, the Z Flag is set to 1 to indicate a zero byte input.

Sign Flag

The Sign (S) Flag stores the state of the most-significant bit of the Accumulator (Bit 7). When the eZ80 performs arithmetic operations on signed numbers, binary two's complement notation is used to represent and process numerical information. A positive number is identified by a 0 in Bit 7. A negative number is identified by a 1. The binary equivalent of the magnitude of a positive number is stored in Bits 0-6 for a total range of 0-127. A negative number is represented by the two's complement of the equivalent positive number. The total range for negative numbers is -1 to -128.

When inputting a byte from an I/O device to a register, **IN r,(C)**, the S Flag indicates either positive (S=0) or negative (S=1) data.

Instructions

The notations in the eZ80 instructions are defined in the following table.

Table 6. Instruction Notations

Mnemonic	Definition
cc	A condition code C, NC, Z, NZ, P, M, PO, PE, V, or NV.
cc'	A condition code C, NC, Z, or NZ.
d	An 8-bit signed displacement -128 to 127.
Mmn	A 16- or 24-bit immediate value or direct address.
m	A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d).
n	An 8-bit immediate value or port number in the range of 0-255 or 0-FFH.
p	A, B, C, D, E, H, L, n, or (HL)
pp	BC, DE, HL, IX+d, or IY+d.
qq	BC, DE, HL, SP, IX, or IY.
r, r'	A, B, C, D, E, H, L.
ii	IX+d or IY+d.
rr	HL, IX, or IY.
s	A, B, C, D, E, H, L, n, (HL), (IX+d), or (IY+d).
ss	BC, DE, HL, or SP.
tt	BC, DE, HL, IX, or IY.
uu	HL, IX+d, or IY+d.
vv	AF, BC, DE, HL, IX, or IY.

ADC A, s

ADD with Carry

Operation

A ← A + s + CY

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand, along with the Carry Flag (c in the F register), is added to the contents of the Accumulator, which contain the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if carry from Bit 3; reset otherwise.
P/V Set if overflow; reset otherwise.
N Reset.
C Set if carry from Bit 7; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
ADC	A,r	X	1	jj
ADC	A,n	X	2	CE, nn
ADC	A,(HL)	X	2	8E
ADC.S	A,(HL)	1	3	52, 8E
ADC.L	A,(HL)	0	3	49, 8E
ADC	A,(IX+d)	X	4	DD, 8E, dd
ADC.S	A,(IX+d)	1	5	52, DD, 8E, dd
ADC.L	A,(IX+d)	0	5	49, DD, 8E, dd
ADC	A,(IY+d)	X	4	FD, 8E, dd
ADC.S	A,(IY+d)	1	5	52, FD, 8E, dd
ADC.L	A,(IY+d)	0	5	49, FD, 8E, dd

jj = binary code **10 001 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

ADC HL, ss

ADD with Carry

Operation

HL ← HL + ss + CY

Description

The ss operand is any of the BC, DE, HL, or SP registers. The ss operand, along with the Carry Flag (C in the F register) is added to the contents of the HL register, which contains the result. ADL mode affects operations with the HL, BC, DE, and SP registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if carry from Bit 11; reset otherwise.
P/V Set if overflow; reset otherwise.
N Reset.
C Set if carry from MSB; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
ADC	HL,ss	X	2	ED, kk
ADC.S	HL,ss	1	3	52, ED, kk
ADC.L	HL,ss	0	3	49, ED, kk

kk = binary code **01 ss1 010** where **ss** identifies the BC, DE, HL, or SP register assembled as follows into the object code.

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

ADD A, s

ADD without Carry

Operation

A ← A + s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is added to the contents of the Accumulator, which contains the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if carry from Bit 3; reset otherwise.
P/V Set if overflow; reset otherwise.
N Reset.
C Set if carry from Bit 7; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
ADD	A,r	X	1	jj
ADD	A,n	X	2	C6, nn
ADD	A,(HL)	X	2	86
ADD.S	A,(HL)	1	3	52, 86
ADD.L	A,(HL)	0	3	49, 86
ADD	A,(IX+d)	X	4	DD, 86, dd
ADD.S	A,(IX+d)	1	5	52, DD, 86, dd
ADD.L	A,(IX+d)	0	5	49, DD, 86, dd
ADD	A,(IY+d)	X	4	FD, 86, dd
ADD.S	A,(IY+d)	1	5	52, FD, 86, dd
ADD.L	A,(IY+d)	0	5	49, FD, 86, dd

jj = binary code **10 000 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

ADD rr, ss

ADD without Carry

Operation

rr ← rr + ss

Description

The ss operand is any of the BC, DE, HL, or SP registers. The destination rr register is any of the HL, IX, or IY registers. The ss operand is added to the contents of the rr register, which contain the result. ADL mode affects operations with the HL, IX, IY, BC, DE, and SP registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Not affected.
H Set if carry from Bit 11; reset otherwise.
P/V Not affected.
N Reset.
C Set if carry from MSB; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
ADD	HL,ss	X	1	kk
ADD.S	HL,ss	1	2	52, kk
ADD.L	HL,ss	0	2	49, kk
ADD	IX,ss	X	2	DD, kk
ADD.S	IX,ss	1	3	52, DD, kk
ADD.L	IX,ss	0	3	49, DD, kk
ADD	IY,ss	X	2	FD, kk
ADD.S	IY,ss	1	3	52, FD, kk
ADD.L	IY,ss	0	3	49, FD, kk

kk = binary code **00 ss1 001** where **ss** identifies the BC, DE, HL, or SP register assembled as follows into the object code.

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

AND A, s

Logical AND

Operation

A ← A • s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is bitwise AND'ed with the contents of the Accumulator, which contain the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set.
P/V Set if parity is even; reset otherwise.
N Reset.
C Reset.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
AND	A,r	X	1	jj
AND	A,n	X	2	E6, nn
AND	A,(HL)	X	2	A6
AND.S	A,(HL)	1	3	52, A6
AND.L	A,(HL)	0	3	49, A6
AND	A,(IX+d)	X	4	DD, A6, dd
AND.S	A,(IX+d)	1	5	52, DD, A6, dd
AND.L	A,(IX+d)	0	5	49, DD, A6, dd
AND	A,(IY+d)	X	4	FD, A6, dd
AND.S	A,(IY+d)	1	5	52, FD, A6, dd
AND.L	A,(IY+d)	0	5	49, FD, A6, dd

jj = binary code **10 100 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

BIT b, m

Test Bit

Operation

Z ← ~mb

Description

The m operand is any of r, (HL), (IX+d), or (IY+d). This instruction tests Bit b in the specified register or memory location and sets the Z Flag if the bit is zero. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if Bit b is zero; reset otherwise.
H Set.
P/V Not affected.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
BIT	b,r	X	2	CB, jj
BIT	b,(HL)	X	3	CB, kk
BIT.S	b,(HL)	1	4	52, CB, kk
BIT.L	b,(HL)	0	4	49, CB, kk
BIT	b,(IX+d)	X	5	DD, CB, dd, kk
BIT.S	b,(IX+d)	1	6	52, DD, CB, dd, kk
BIT.L	b,(IX+d)	0	6	49, DD, CB, dd, kk
BIT	b,(IY+d)	X	5	FD, CB, dd, kk
BIT.S	b,(IY+d)	1	6	52, FD, CB, dd, kk
BIT.L	b,(IY+d)	0	6	49, FD, CB, dd, kk

jj = binary code **01 bbb rrr**, and **kk** = binary code **01 bbb 110**; where **rrr** identifies the A, B, C, D, E, H, or L register and **bbb** identifies the bit tested assembled as follows into the object code.

Bit Tested	bbb	Register	rrr
0	**000**	A	**111**
1	**001**	B	**000**
2	**010**	C	**001**
3	**011**	D	**010**
4	**100**	E	**011**
5	**101**	H	**100**
6	**110**	L	**101**
7	**111**		

CALL Mmn

CALL Subroutine

Operation

(SP) ← PC
PC ← Mmn

Description

The return address, which follows this instruction, is pushed onto the stack and then the program counter is loaded with the Mmn operand and execution continues at that address. The Mmn operand is a 16- or 24-bit address, depending on the instruction and/or the ADL mode. The low-order byte of the mn operand is the first byte after the Op Code.

ADL	Prefix	Operation
0	None	Stack 2-byte logical return address using SPS mapped by MBASE. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	None	Stack 3-byte return address using SPL. Keep ADL set to 1. Load a 3-byte logical address from the instruction into PC.
0	.IS	Stack 2-byte logical return address using SPS mapped by MBASE. Stack a **00** byte using SPL. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	.IS	Stack 2 LS bytes of the return address using SPS mapped by MBASE. Stack the MS byte of the return address using SPL. Stack a **01** byte using SPL. Clear ADL to 0. Load a 2-byte logical address from the instruction into PC.
0	.IL	Stack 2-byte logical return address using SPL. Stack a **00** byte using SPL. Set ADL to 1. Load a 3-byte address from the instruction into PC.
1	.IL	Stack 3-byte return address using SPL. Stack a **01** byte using SPL. Keep ADL set to 1. Load a 3-byte address from the instruction into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CALL	mn	0	5	CD, mn, mn
CALL	Mmn	1	7	CD, mn, mn, MM
CALL.IS	mn	0	7	40, CD, mn, mn
CALL.IS	mn	1	8	49, CD, mn, mn
CALL.IL	Mmn	0	8	52, CD, mn, mn, MM
CALL.IL	Mmn	1	9	5B, CD, mn, mn, MM

CALL cc, Mmn

Conditional CALL Subroutine

Operation

If cc True:
(SP) ← PC
PC ← Mmn

Description

If condition cc is true, the return address, which follows this instruction, is pushed onto the stack and then the program counter is loaded with the Mmn operand and execution continues at the new PC address. The Mmn operand is a 16- or 24-bit address, depending on the instruction and/or the ADL mode. The low-order byte of the mn operand is the first byte after the Op Code.

ADL	Prefix	Operation
0	None	Stack 2-byte logical return address using SPS mapped by MBASE. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	None	Stack 3-byte return address using SPL. Keep ADL set to 1. Load a 3-byte logical address from the instruction into PC.
0	.IS	Stack 2-byte logical return address using SPS mapped by MBASE. Stack a **00** byte using SPL. Keep ADL cleared to 0. Load a 2-byte logical address from the instruction into PC.
1	.IS	Stack two LS bytes of the return address using SPS mapped by MBASE. Stack the MS byte of the return address using SPL. Stack a **01** byte using SPL. Clear ADL to 0. Load a 2-byte logical address from the instruction into PC.
0	.IL	Stack 2-byte logical return address using SPL. Stack a **00** byte using SPL. Set ADL to 1. Load a 3-byte address from the instruction into PC.
1	.IL	Stack 3-byte return address using SPL. Stack a 01 byte using SPL. Keep ADL set to 1. Load a 3-byte address from the instruction into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CALL	cc,mn	0	5	kk, mn, mn
CALL	cc,Mmn	1	7	kk, mn, mn, MM
CALL.IS	cc,mn	0	7	40, kk, mn, mn
CALL.IS	cc,mn	1	8	49, kk, mn, mn
CALL.IL	cc,Mmn	0	8	52, kk, mn, mn, MM
CALL.IL	cc,Mmn	1	9	5B, kk, mn, mn, MM

kk = binary code **11 ccc 100** where **ccc** identifies one of status conditions assembled as follows into the object code.

ccc	Condition	Relevant Flag
000	NZ (non zero)	Z
001	Z (zero)	Z
010	NC (non carry)	C
011	C (carry)	C
100	PO (parity odd)	P/V
101	PE (parity even)	P/V
110	P (sign positive)	S
111	M (sign negative)	s

CCF

Complement Carry Flag

Operation

CY ← ~CY

Description

The Carry Flag bit in the F register is inverted.

Condition Bits Affected

S Not affected.
Z Not affected.
H Previous carry is copied.
P/V Not affected.
N Reset.
C Set if carry was cleared to 0 before operation; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CCF	—	X	1	3F

CP A, s

Compare with Accumulator

Operation

A - s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is compared with (subtracted from) the contents of the Accumulator. The execution of this instruction does not affect the contents of the Accumulator or the s operand. ADL mode affects operation with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if overflow: reset otherwise.
N Set.
C Set if borrow; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CP	A,r	X	1	jj
CP	A,n	X	2	FE, nn
CP	A,(HL)	X	2	BE
CP.S	A,(HL)	1	3	52, BE
CP.L	A,(HL)	0	3	49, BE
CP	A,(IX+d)	X	4	DD, BE, dd
CP.S	A,(IX+d)	1	5	52, DD, BE, dd
CP.L	A,(IX+d)	0	5	49, DD, BE, dd
CP	A,(IY+d)	X	4	FD, BE, dd
CP.S	A,(IY+d)	1	5	52, FD, BE, dd
CP.L	A,(IY+d)	0	5	49, FD, BE, dd

jj = binary code **10 111 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

CPD

Compare Decrement

Operation

A – (HL)
HL ← HL – 1
BC ← BC – 1

Description

The contents of the memory location that the HL register points to are compared to the contents of the Accumulator, which are not affected by this instruction. The HL and BC registers are decremented.

ADL mode affects operation with the HL and BC registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if A = (HL); reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if BC –1 ≠ **0**; reset otherwise.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CPD	—	X	3	ED, A9
CPD.S	—	1	4	52, ED, A9
CPD.L	—	0	4	49, ED, A9

CPDR

Compare Decrement Repeat

Operation

repeat :
{ A – (HL)
HL ← HL – 1
BC ← BC – 1
} while (not Z and BC ≠ 0)

Description

The contents of the memory location that the HL register points to are compared to the contents of the Accumulator, which are not affected by this instruction. The HL and BC registers are decremented. The above operation is repeated, unless
A = (HL) or BC are decremented to zero.

ADL mode affects operation with the HL and BC registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if A = (HL); reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if BC –1 ≠ 0; reset otherwise.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CPDR	—	X	4/3	ED, B9
CPDR.S	—	1	5/3	52, ED, B9
CPDR.L	—	0	5/3	49, ED, B9

CPI

Compare Increment

Operation

A – (HL)
HL ← HL + 1
BC ← BC – 1

Description

The contents of the memory location that the HL register points to are compared to the contents of the Accumulator, which are not affected by this instruction. The HL register increments and the BC register decrements.

ADL mode affects operation with the HL and BC registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if A = (HL); reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if BC –1 ≠ 0; reset otherwise.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CPI	—	X	3	ED, A1
CPI.S	—	1	4	52, ED, A1
CPI.L	—	0	4	49, ED, A1

CPIR

Compare Increment Repeat

Operation

repeat :
{ A – (HL)
HL ← HL + 1
BC ← BC – 1
} while (not Z and BC ≠ 0)

Description

The contents of the memory location that the HL register points to are compared to the contents of the Accumulator, which are not affected by this instruction. The HL register increments and the BC register decrements. The above operation is repeated, unless A = (HL) or BC decrements to zero.

ADL mode affects operation with the HL and BC registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if A = (HL); reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if BC –1 ≠ 0; reset otherwise.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CPIR	—	X	4/3	ED, B1
CPIR.S	—	1	5/3	52, ED, B1
CPIR.L	—	0	5/3	49, ED, B1

CPL

Complement Accumulator

Operation

A ← ~A

Description

All bits in the Accumulator (register A) are inverted (1's complement).

Condition Bits Affected

S Not affected.
Z Not affected.
H Set.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
CPL	—	X	1	2F

DAA

Decimal Adjust Accumulator

Operation

—

Description

This instruction conditionally adjusts the Accumulator for BCD addition and subtraction operations. For addition (ADD, ADC, INC) or subtraction (SUB, SBC, DEC, NEG), the following table indicates the operation performed.

Operation	C Before DAA	Hex Value in Upper Digit (Bits 7-4)	H Before DAA	Hex Value in Lower Digit (Bits 3-0)	Number Added to Byte	C After DAA
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
ADD	0	A-F	0	0-9	60	1
ADC	0	9-F	0	A-F	66	1
INC	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB	0	0-9	0	0-9	00	0
SBC	0	0-8	1	6-F	FA	0
DEC	1	7-F	0	0-9	A0	1
NEG	1	6-F	1	6-F	9A	1

Condition Bits Affected

S Set if the MSB of the Accumulator is 1 after the operation; reset otherwise.
Z Set if result is zero; reset otherwise.
H See instruction.
P/V Set if the Accumulator is even parity after the operation; reset otherwise.
N Not affected.
C See instruction.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
DAA	—	X	1	27

DEC qq

Decrement

Operation

If .L or (ADL and not .S)
qq[23:0] ← qq[23:0] – 1
else
qq[15:0] ← qq[15:0] – 1
qq[23:16] ← 0

Description

The qq operand is any of the BC, DE, HL, IX, IY, or SP registers. The contents of the specified register decrement by 1. ADL mode affects operation with the BC, DE, HL, SP, IX and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
DEC	ss	X	1	kk
DEC.S	ss	1	2	52, kk
DEC.L	ss	0	2	49, kk
DEC	IX	X	1	DD, 2B
DEC.S	IX	1	2	52, DD, 2B
DEC.L	IX	0	2	49, DD, 2B
DEC	IY	X	1	FD, 2B
DEC.S	IY	1	2	52, FD, 2B
DEC.L	IY	0	2	49, FD, 2B

kk = binary code **00 ss1 011** where **ss** identifies the BC, DE, HL, or SP register assembled as follows into the object code.

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

DEC m

Decrement

Operation

m ← m – 1

Description:

The m operand is any of r, (HL), (IX+d), or (IY+d). The m operand decrements by 1. ADL mode affects operation with registers HL, IX, and IY. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set is borrowed from Bit 4; reset otherwise.
P/V Set if operand was 80H before operation; reset otherwise.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
DEC	r	X	1	jj
DEC	(HL)	X	4	35
DEC.S	(HL)	1	5	52, 35
DEC.L	(HL)	0	5	49, 35
DEC	(IX+d)	X	6	DD, 35, dd
DEC.S	(IX+d)	1	7	52, DD, 35, dd
DEC.L	(IX+d)	0	7	49, DD, 35, dd
DEC	(IY+d)	X	6	FD, 35, dd
DEC.S	(IY+d)	1	7	52, FD, 35, dd
DEC.L	(IY+d)	0	7	49, FD, 35, dd

jj = binary code **00 rrr 101** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

DI

Disable Interrupt

Operation

IEF ← 0

Description:

This instruction disables the maskable interrupts by resetting the interrupt enable flags (IEF1 and IEF2).

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
DI	—	X	1	F3

DJNZ d

Decrement B Jump not Zero

Operation

B ← B – 1
If B ≠ 0
PC ← PC + d

Description

The B register decrements by 1. If the resultant value in register B is not zero, the two's complement displacement d is added to the value of the program counter. The jump is measured from the address of the instruction Op Code following this instruction.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
DJNZ	d	X	2/4	10, dd

EI

Enable Interrupt

Operation

IEF ← 1

Description

This instruction sets the interrupt enable flags (IEF1 and IEF2) to a logic 1, which allows any maskable interrupt to be recognized.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
EI	—	X	1	FB

EX AF, AF’

Exchange AF and AF’

Operation

A ↔ A’
F ↔ F’

Description

The contents of the Accumulator and Flag registers (AF) are exchanged with the contents of the alternate Accumulator and alternate Flag registers (AF’).

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
EX	AF,AF’	X	1	08

EX DE, HL

Exchange DE with HL

Operation

DE ↔ HL

Description

The contents of the DE register are exchanged with the contents of the HL register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
EX	DE,HL	X	1	EB

EX (SP), rr

Exchange Stack and Register

Operation

If .L or (ADL and not .S)
(SPL+2) ↔ rr[23:16]
(SPL+1) ↔ rr[15:8]
(SPL+0) ↔ rr[7:0]
else
(SPS+1) ↔ rr[15:8]
(SPS+0) ↔ rr[7:0]
rr[23:16] ← 0

Description

The contents of the specified register are exchanged with the memory location specified by the contents of the stack pointer.

ADL mode affects operation with the SP, HL, IX, and IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
EX	(SP),HL	X	5,7	E3
EX.S	(SP),HL	1	6	52, E3
EX.L	(SP),HL	0	8	49, E3
EX	(SP),IX	X	6,8	DD, E3
EX.S	(SP),IX	1	7	52, DD, E3
EX.L	(SP),IX	0	9	49, DD, E3
EX	(SP),IY	X	6,8	FD, E3
EX.S	(SP),IY	1	7	52, FD, E3
EX.L	(SP),IY	0	9	49, FD, E3

EXX

Exchange with Alternate Register Set

Operation

BC ↔ BC'
DE ↔ DE'
HL ↔ HL'

Description

The contents of the BC, DE, and HL registers are exchanged with the alternate BC, DE, and HL registers.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
EXX	—	X	1	D9

HALT

Halt

Operation

—

Description

The **HALT** instruction suspends CPU operation until a subsequent interrupt or reset is received. While in the halt state, the CPU executes NOPs.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
HALT	—	X	1	76

IM n

Set Interrupt Mode

Operation

—

Description

The n operand is any of the following interrupt modes:

- Interrupt mode 0—In this mode, the interrupting device inserts an instruction on the data bus during an interrupt acknowledge cycle.
- Interrupt mode 1—In this mode, the CPU responds to an interrupt by executing a restart to location 000038H.
- Interrupt mode 2—In this mode, the interrupting device places the low-order address of the interrupt vector on the data bus during an interrupt acknowledge cycle. The I register provides the high-order byte of the interrupt vector table address. The 16-bit value at this address is the starting address of the interrupt service routine.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IM	0	X	2	ED, 46
IM	1	X	2	ED, 56
IM	2	X	2	ED, 5e

IN A (n)

Input I/O Port to Accumulator

Operation

A ← (n)

Description

The n operand is placed on address bus (7:0), the contents of the Accumulator on address bus (15:8), and zero on address bus (23:16). The byte at this I/O address is loaded into the Accumulator.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IN	(n)	X	3	DB, nn

IN r, (C)

Input I/O Port to Register

Operation

r ← (C)

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is loaded into the specified register.

Condition Bits Affected

S Set if Bit 7 is 1; reset otherwise.
Z Set if byte is 0; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IN	(C)	ind	3	ED, jj

jj = binary code **01 rrr 000** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

IN0 r, (n)

Input I/O Port to Register

Operation

adr[23:8] ← 0
adr[7:0] ← n
r ← (adr)

Description

The n operand is placed on address bus (7:0) and address bus (23:8) are zero. The byte at this I/O address is loaded into the specified register.

Condition Bits Affected

S Set if Bit 7 is 1; reset otherwise.
Z Set if byte is 0; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IN0	r,(n)	X	4	ED, jj, nn

jj = binary code **00 rrr 000** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

INC qq

Increment Register

Operation

If .L or (ADL and not .S)
qq[23:0] ← qq[23:0] + 1
else
qq[15:0] ← qq[15:0] + 1
qq[23:16] ← 0

Description

The qq operand is any of the BC, DE, HL, SP, IX, or IY registers. The contents of the specified register increment by 1. ADL mode affects operation with the BC, DE, HL, SP, IX and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INC	ss	X	1	kk
INC.S	ss	1	2	52, kk
INC.L	ss	0	2	49, kk
INC	IX	X	1	DD, 23
INC.S	IX	1	2	52, DD, 23
INC.L	IX	0	2	49, DD, 23
INC	IY	X	1	FD, 23
INC.S	IY	1	2	52, FD, 23
INC.L	IY	0	2	49, FD, 23

kk = binary code **00 ss0 011** where **ss** identifies the BC, DE, HL, or SP register assembled as follows into the object code.

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

INC m

Increment

Operation

m ← m + 1

Description

The m operand is any of r, (HL), (IX+d), or (IY+d). The m operand increments by 1. ADL mode affects operation with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if carry from Bit 3.
P/V Set if operand was 7FH before operation.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INC	r	X	1	jj
INC	(HL)	X	4	34
INC.S	(HL)	1	5	52, 34
INC.L	(HL)	0	5	49, 34
INC	(IX+d)	X	6	DD, 34, dd
INC.S	(IX+d)	1	7	52, DD, 34, dd
INC.L	(IX+d)	0	7	49, DD, 34, dd
INC	(IY+d)	X	6	FD, 34, dd
INC.S	(IY+d)	1	7	52, FD, 34, dd
INC.L	(IY+d)	0	7	49, FD, 34, dd

jj = binary code **00 rrr 100** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code.

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

IND

Input and Decrement HL

Operation

(HL) ← (C)
B ← B – 1
HL ← HL – 1

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and HL registers are decremented. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IND	—	X	5	ED, AA
IND.S	—	1	6	52, ED, AA
IND.L	—	0	6	49, ED, AA

IND2

Input and Decrement HL

Operation

(HL) ← (C)
B ← B – 1
C ← C – 1
HL ← HL – 1

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B, C, and HL registers are decremented. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IND2	—	x	5	ED, 8C
IND2.S	—	1	6	52, ED, 8C
IND2.L	—	0	6	49, ED, 8C

IND2R

Input and Decrement HL - Repeat

Operation

repeat:
{ (HL) ← (C)
B ← B – 1
C ← C – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B, C, and HL registers are decremented. The instruction repeats until the B register equals zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
IND2R	—	X	5/3	ED, 9C
IND2R.S	—	1	6/3	52, ED, 9C
IND2R.L	—	0	6/3	49, ED, 9C

INDM

Input and Decrement HL

Operation

(HL) ← (C)
B ← B – 1
C ← C – 1
HL ← HL – 1

Description

The contents of the C register are placed on address bus (7:0) and zero on address bus (23:8). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B, C, and HL registers are decremented. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INDM	—	X	5	ED, 8A
INDM.S	—	1	6	52, ED, 8A
INDM.L	—	0	6	49, ED, 8A

INDMR

Input and Decrement HL - Repeat

Operation

repeat:
{ (HL) ← (0C)
B ← B – 1
C ← C – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of the C register are placed on address bus (7:0) and zero on address bus (23:8). The byte at this I/O address is read into the CPU. The contents of HL are placed on the address bus and the byte is written to the memory address specified by the HL register. The B, C, and HL registers are decremented. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INDMR	—	X	5/3	ED, 9A
INDMR.S	—	1	6/3	52, ED, 9A
INDMR.L	—	0	6/3	49, ED, 9A

INDR

Input and Decrement HL - Repeat

Operation

repeat:
{ (HL) ← (C)
B ← B – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and HL registers are decremented. The instruction repeats until the B register equals zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INDR	—	X	5/3	ED, BA
INDR.S	—	1	6/3	52, ED, BA
INDR.L	—	0	6/3	49, ED, BA

INI

Input and Increment HL

Operation

(HL) ← (C)
B ← B – 1
HL ← HL + 1

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B register decrements and the HL register increments. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INI	—	X	5	ED, A2
INI.S	—	1	6	52, ED, A2
INI.L	—	0	6	49, ED, A2

INI2

Input and Increment HL

Operation

(HL) ← (C)
B ← B – 1
C ← C – 1
HL ← HL + 1

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and C registers are decremented and the HL register increments. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INI2	—	—	5	ED, 84
INI2.S	—	—	6	52, ED, 84
INI2.L	—	—	6	49, ED, 84

INI2R

Input and Increment HL - Repeat

Operation

repeat:
{ (HL) ← (C)
B ← B – 1
C ← C – 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and C registers are decremented and the HL register increments. The instruction repeats until the B register equals zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INI2R	—	X	5/3	ED, 94
INI2R.S	—	1	6/3	52, ED, 94
INI2R.L	—	0	6/3	49, ED, 94

INIM

Input and Increment HL

Operation

(HL) ← (0C)
B ← B – 1
C ← C – 1
HL ← HL + 1

Description

The contents of the C register are placed on address bus (7:0) and zero on address bus (23:8). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and C registers are decremented and the HL register increments. The Z Flag is set to 1 if the B register decrements to zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B-1 = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INIM	—	X	5	ED, 82
INIM.S	—	1	6	52, ED, 82
INIM.L	—	0	6	49, ED, 82

INIMR

Input and Increment HL - Repeat

Operation

repeat:
{ (HL) ← (0C)
B ← B – 1
C ← C – 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of the C register are placed on address bus (7:0) and zero on address bus (23:8). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B and C registers are decremented and the HL register increments. The instruction repeats until the B register equals zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INIMR	—	X	6/3	ED, 92
INIMR.S	—	1	6/3	52, ED, 92
INIMR.L	—	0	6/3	49, ED, 92

INIR

Input and Increment HL - Repeat

Operation

repeat:
{ (HL) ← (C)
B ← B – 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of BC (15:0) are placed on address bus (15:0) and zero on address bus (23:16). The byte at this I/O address is read into the CPU. The contents of HL are then placed on the address bus and the byte is written to the memory address specified by the HL register. The B register decrements and the HL register increments. The instruction repeats until the B register equals zero.

ADL mode affects operation with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
INIR	—	X	5/3	ED, B2
INIR.S	—	1	6/3	52, ED, B2
INIR.L	—	0	6/3	49, ED, B2

JP (rr)
Jump Indirect

Operation

```
If .S
        ADL ← 0
else if .L
        ADL ← 1
If ADL=1
        PC[23:0] ← rr[23:0]
else
        PC[15:0] ← rr[15:0]
        PC[23:16] ← 0
```

Description

The program counter is loaded with the contents of the specified register.

ADL	Prefix	Operation
0	None	Load a 2-byte logical address from the register into PC. Keep ADL cleared to 0.
1	None	Load a 3-byte logical address from the register into PC. Keep ADL set to 1.
X	.L16	Clear ADL. Load a 3-byte logical address from the register into PC.
X	.S15	Set ADL to 1. Load a 2-byte logical address from the register into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
		0		
JP	(HL)	1	3	E9
JP.S15	(HL)	X	4	40, E9
JP.L16	(HL)	X	4	5B, E9
		0		
JP	(IX)	1	4	DD, E9
JP.S15	(IX)	X	5	40, DD, E9
JP.L16	(IX)	X	5	5B, DD, E9
		0		
JP	(IY)	1	4	FD, E9
JP.S15	(IY)	X	5	40, FD, E9
JP.L16	(IY)	X	5	5B, FD, E9

JP Mmn

Jump

Operation

```
If .IS
    ADL ← 0
else if .IL
    ADL ← 1

If ADL=1
    PC[23:0] ← Mmn[23:0]
else
    PC[15:0] ← mn[15:0]
    PC[23:16] ← 0
```

Description

The program counter is loaded with the instruction operand. The first byte after the Op Code is the low-order byte of the operand.

ADL	Prefix	Operation
0	None	Load a 2-byte logical address from the instruction into PC. Keep ADL cleared to 0.
1	None	Load a 3-byte logical address from the instruction into PC. Keep ADL set to 1.
X	.L15	Clear ADL to 0. Load a 3-byte logical address from the instruction into PC.
X	.S16	Set ADL to 1. Load a 2-byte logical address from the instruction into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
JP	mn	0	4	C3, mn, mn
JP	Mmn	1	5	C3, mn, mn, MM
JP.S15	mn	X	5	40, C3, mn, mn
JP.L16	Mmn	X	6	5B, C3, mn, mn, MM

JP cc, Mmn

Conditional Jump

Operation

```
If cc TRUE:
If .IS
    ADL ← 0
else if .IL
    ADL ← 1
If ADL=1
    PC[23:0] ← Mmn[23:0]
else
    PC[15:0] ← mn[15:0]
    PC[23:16] ← 0
```

Description

If the condition is true, then the program counter is loaded with the instruction operand. The first byte after the Op Code is the low-order byte of the operand.

ADL	Prefix	Operation
0	None	Load a 2-byte logical address from the instruction into PC. Keep ADL cleared to 0.
1	None	Load a 3-byte logical address from the instruction into PC. Keep ADL set to 1.
X	.LIL	Clear ADL to 0. Load a 3-byte logical address from the instruction into PC.
X	.SIS	Set ADL to 1. Load a 2-byte logical address from the instruction into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
JP	cc,mn	0	4	kk, mn, mn
JP	cc,Mmn	1	5	kk, mn, mn, MM
JP.SS	cc,mn	X	5	40, kk, mn, mn
JP.LIL	cc,Mmn	X	6	5B, kk, mn, mn, MM

kk = binary code **11 ccc 010** where **ccc** identifies one of status conditions assembled as follows into the object code.

ccc	Condition	Relevant Flag
000	NZ (non-zero)	Z
001	Z (zero)	Z
010	NC (non-carry)	C
011	C (carry)	C
100	PO (parity odd)	P/V
101	PE (parity even)	P/V
110	P (sign positive)	S
111	M (sign negative)	s

JR d

Jump Relative

Operation

PC ← PC + d

Description

The two's-complement displacement d is added to the program counter. The jump is measured from the address of the byte following the instruction (JR Op Code +2).

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
JR	d	X	3	18, dd

JR cc, d

Conditional Jump Relative

Operation

If cc' TRUE:
PC ← PC + d

Description

If the condition is true, then the two's-complement displacement d is added to the program counter. The jump is measured from the address of the byte following the instruction (JR Op Code +2).

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
JR	cc,d	X	2/3	kk, dd

kk = binary code **00 ccc 000** where **ccc** identifies one of status conditions assembled as follows into the object code.

ccc	Condition	Relevant Flag
100	NZ (non-zero)	Z
101	Z (zero)	Z
110	NC (non-carry)	C
111	C (carry)	C

LD A, I

Load

Operation

A ← I

Description

The contents of the Interrupt Vector register are loaded into the Accumulator.

Condition Bits Affected

S Set if the I register is negative; reset otherwise.
Z Set if the I register is zero; reset otherwise.
H Reset.
P/V Contains contents of IEF2.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A, I	X	2	ED, 57

LD A, MB

Load

Operation

A ← MBASE

Description

The contents of the Memory Base register are loaded into the Accumulator. In non-ADL mode, no operation occurs (two cycle NOP).

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,MB	1	2	ED, 6E

LD A, (Mmn)

Load

Operation

A ← (Mmn)

Description

The contents of the specified memory location are loaded into the Accumulator.

The Mmn operand is 16-bit if non-ADL or the .SIS suffix is used. The Mmn operand is 24-bit if ADL mode or the .LIL suffix is used.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,(mn)	0	4	3A, nn, mm
LD	A,(Mmn)	1	5	3A, nn, mm, MM
LD.SIS	A,(mn)	1	5	40, 3A, nn, mm
LD.LIL	A,(Mmn)	0	6	5B, 3A, nn, mm, MM

LD A, (pp)

Load

Operation

A ← (pp)

Description

The pp operand is any of BC, DE, HL, IX+d, or IY+d. The contents of the memory location specified by the contents of the specified register are loaded into the Accumulator.

ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,(BC)	X	2	0A
LD.S	A,(BC)	1	3	52, 0A
LD.L	A,(BC)	0	3	49, 0A
LD	A,(DE)	X	2	1A
LD.S	A,(DE)	1	3	52, 1A
LD.L	A,(DE)	0	3	49, 1A
LD	A,(HL)	X	—	7E
LD.S	A,(HL)	1	—	52, 7E
LD.L	A,(HL)	0	—	40, 7E
LD	A,(IX+d)	X	—	DD, 7E, dd
LD.S	A,(IX+d)	1	—	52, DD, 7E, dd
LD.L	A,(IX+d)	0	—	49, DD, 7E, dd
LD	A,(IX+d)	X	2	FD, 7E, dd
LD.S	A,(IX+d)	1	3	52, FD, 7E, dd
LD.L	A,(IX+d)	0	3	49, FD, 7E, dd

LD A, R

Load

Operation

A ← R

Description

The contents of the Refresh register are loaded into the Accumulator.

Condition Bits Affected

S Set if the R register is negative; reset otherwise.
Z Set if the R register is zero; reset otherwise.
H Reset.
P/V Contains contents of IEF2.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,I	X	2	ED, 5F

LD (HL), tt

Load

Operation

(HL) ← tt

Description

The tt operand is any of BC, DE, HL, IX, or IY. The contents of the multi-byte tt register, which remains unchanged, are stored into the memory location specified by the contents of the multi-byte HL register.

The length of the multi-byte HL and qq registers may be a 16- or 24-bit value, depending on the ADL mode. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(HL),BC	0/1	4/5	ED, 0F
LD.S	(HL),BC	1	5	52, ED, 0F
LD.L	(HL),BC	0	6	49, ED, 0F
LD	(HL),DE	0/1	4/5	ED, 1F,
LD.S	(HL),DE	1	5	52, ED, 1F
LD.L	(HL),DE	0	6	49, ED, 1F
LD	(HL),HL	0/1	4/5	ED, 2F
LD.S	(HL),HL	1	5	52, ED, 2F
LD.L	(HL),HL	0	6	49, ED, 2F
LD	(HL),IX	0/1	4/5	ED, 3F
LD.S	(HL),IX	1	5	52, ED, 3F
LD.L	(HL),IX	0	6	49, ED, 3F
LD	(HL),IY	0/1	4/5	ED, 3E
LD.S	(HL),IY	1	5	52, ED, 3E
LD.L	(HL),IY	0	6	49, ED, 3E

LD I, A

Load

Operation

I ← A

Description

The contents of the Accumulator are loaded into the Interrupt Vector register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	I,A	X	2	ED, 47

LD (ii), tt

Load

Operation

(ii) ← tt

Description

The tt operand is any of BC, DE, HL, IX, or IY. The ii operand is any of IX+d or IY+d. The contents of the multi-byte tt register, which remain unchanged, are stored into the memory location specified by the contents of the multi-byte ii register. The length of the multi-byte ii and tt registers may be a 16- or 24-bit value, depending on the ADL mode. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(IX+d),BC	0/1	5/6	DD, 0F, dd
LD.S	(IX+d),BC	1	6	52, DD, 0F, dd
LD.L	(IX+d),BC	0	7	49, DD, 0F, dd
LD	(IX+d),DE	0/1	5/6	DD, 1F, dd
LD.S	(IX+d),DE	1	6	52, DD, 1F, dd
LD.L	(IX+d),DE	0	7	49, DD, 1F, dd
LD	(IX+d),HL	0/1	5/6	DD, 2F, dd
LD.S	(IX+d),HL	1	6	52, DD, 2F, dd
LD.L	(IX+d),HL	0	7	49, DD, 2F, dd
LD	(IX+d),IX	0/1	5/6	DD, 3F, dd
LD.S	(IX+d),IX	1	6	52, DD, 3F, dd
LD.L	(IX+d),IX	0	7	49, DD, 3F, dd
LD	(IX+d),IY	0/1	5/6	DD, 3E, dd
LD.S	(IX+d),IY	1	6	52, DD, 3E, dd
LD.L	(IX+d),IY	0	7	49, DD, 3E, dd
LD	(IY+d),BC	0/1	5/6	FD, 0F, dd
LD.S	(IY+d),BC	1	6	52, FD, 0F, dd
LD.L	(IY+d),BC	0	7	49, FD, 0F, dd
LD	(IY+d),DE	0/1	5/6	FD, 1F, dd
LD.S	(IY+d),DE	1	6	52, FD, 1F, dd
LD.L	(IY+d),DE	0	7	49, FD, 1F, dd
LD	(IY+d),HL	0/1	5/6	FD, 2F, dd
LD.S	(IY+d),HL	1	6	52, FD, 2F, dd
LD.L	(IY+d),HL	0	7	49, FD, 2F, dd
LD	(IY+d),IX	0/1	5/6	FD, 3E, dd
LD.S	(IY+d),IX	1	6	52, FD, 3E, dd
LD.L	(IY+d),IX	0	7	49, FD, 3E, dd
LD	(IY+d),IY	0/1	5/6	FD, 3F, dd
LD.S	(IY+d),IY	1	6	52, FD, 3F, dd
LD.L	(IY+d),IY	0	7	49, FD, 3F, dd

LD MB, A

Load

Operation

MBASE ← A

Description

In ADL mode, the contents of the Accumulator are loaded into the MBASE register. Otherwise, no operation occurs.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,I	X	2	ED, 6E

LD (Mmn), A

Load

Operation

(Mmn) ← A

Description

The contents of the Accumulator are stored into the specified memory location, but remain unchanged.

The Mmn operand is 16-bit if non-ADL or the .SIS suffix is used. The Mmn operand is 24-bit if ADL mode or the .LIL suffix is used.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(mn),A	0	3	32, nn, mm
LD	(Mmn),A	1	4	32, nn, mm, MM
LD.SIS	(mn),A	1	4	40, 32, nn, mm
LD.LIL	(Mmn),A	0	5	5B, 32, nn, mm, MM

LD (Mmn), qq

Load

Operation

(Mmn) ← qq

Description

The qq operand is any of BC, DE, HL, SP, IX, or IY. The contents of the multi-byte qq register are stored in the specified memory location, but remain unchanged. If ADL mode is set to 1, the entire 24-bit address is specified in the direct address operand and the 24-bit register is loaded into the specified address locations (the direct address and the next two locations). If ADL is reset, the address is generated with MBASE and the 16-bit direct address operand and the 16-bit register is loaded into the specified address locations (the direct address and the next location). SPL is used in ADL mode and SPS is used in non-ADL mode. The Mmn operand and the length of the multi-byte qq register may be a 16- or 24-bit operand, depending on the ADL mode. ADL mode may be overridden with the .SIS or .LIL suffix.

Condition Bits Affected

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(mn),BC	0	6	ED, 43, nn, mm
LD	(Mmn),BC	1	8	ED, 43 ,nn, mm, MM
LD.SIS	(mn),BC	1	7	40, ED, 43, nn, mm
LD.LIL	(Mmn),BC	0	9	5B, ED, 43, nn, mm, MM
LD	(mn),DE	0	6	ED, 53, nn, mm
LD	(Mmn),DE	1	8	ED, 53 ,nn, mm, MM
LD.SIS	(mn),DE	1	7	40, ED, 53, nn, mm
LD.LIL	(Mmn),DE	0	9	5B, ED, 53, nn, mm, MM
LD	(mn),HL	0	5	22, nn, mm
LD	(Mmn),HL	1	7	22 ,nn, mm, MM
LD.SIS	(mn),HL	1	6	40, 22, nn, mm
LD.LIL	(Mmn),HL	0	8	5B, 22, nn, mm, MM
LD	(mn),SP	0	6	ED, 73, nn, mm
LD	(Mmn),SP	1	8	ED, 73 ,nn, mm, MM
LD.SIS	(mn),SP	1	7	40, ED, 73, nn, mm
LD.LIL	(Mmn),SP	0	9	5B, ED, 73, nn, mm, MM
LD	(mn),IX	0	6	DD, 22, nn, mm
LD	(Mmn),IX	1	8	DD, 22 ,nn, mm, MM
LD.SIS	(mn),IX	1	7	40, DD, 22, nn, mm
LD.LIL	(Mmn),IX	0	9	5B, DD, 22, nn, mm, MM
LD	(mn),IY	0	6	FD, 22, nn, mm
LD	(Mmn),IY	1	8	FD, 22 ,nn, mm, MM
LD.SIS	(mn),IY	1	7	40, FD, 22, nn, mm
LD.LIL	(Mmn),IY	0	9	5B, FD, 22, nn, mm, MM

LD (pp), A

Load

Operation

(pp) ← A

Description

The pp operand is any of BC, DE, HL, IX+d, or IY+d. The contents of the Accumulator are stored in the specified memory location, but remain unchanged. ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(BC),A	X	2	02
LD.S	(BC),A	1	3	52, 02
LD.L	(BC),A	0	3	49, 02
LD	(DE),A	X	2	12
LD.S	(DE),A	1	3	52, 12
LD.L	(DE),A	0	3	49, 12
LD	(HL),A	X	2	77
LD.S	(HL),A	1	3	52, 77
LD.L	(HL),A	0	3	40, 77
LD	(IX+d),A	X	4	DD, 77, dd
LD.S	(IX+d),A	1	5	52, DD, 77, dd
LD.L	(IX+d),A	0	5	49, DD, 77, dd
LD	(IX+d),A	X	4	FD, 77, dd
LD.S	(IX+d),A	1	5	52, FD, 77, dd
LD.L	(IX+d),A	0	5	49, FD, 77, dd

LD qq, Mmn

Load

Operation

qq ← Mmn

Description

The qq operand is any of BC, DE, HL, SP, IX, or IY. The immediate Mmn operand is loaded into the multi-byte qq register. The SP register is special in that SPL is used in ADL mode and SPS is used in non-ADL mode.

ADL mode affects operations with the BC, DE, HL, SP, IX, or IY register . ADL mode may be overridden with the .SIS or .LIL suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	ss,mn	0	3	kk, nn, mm
LD	ss,Mmn	1	4	kk, nn, mm, MM
LD.LIL	ss,Mmn	0	5	5B, kk, nn, mm, MM
LD.SIS	ss,mn	1	4	40, kk, nn, mm
LD	IX,mn	0	4	DD, 21, nn, mm
LD	IX,Mmn	1	5	DD, 21, nn, mm, MM
LD.LIL	IX,Mmn	0	6	5B, DD, 21, nn, mm, MM
LD.SIS	IX,mn	1	5	40, DD, 21, nn, mm
LD	IY,mn	0	4	FD, 21, nn, mm
LD	IY,Mmn	1	5	FD, 21, nn, mm, MM
LD.LIL	IY,Mmn	0	6	5B, FD, 21, nn, mm, MM
LD.SIS	IY,mn	1	5	40, FD, 21, nn, mm

kk = binary code **00 ss0 001** where **ss** identifies the BC, DE, HL, or SP register assembled as follows in the object code:

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

LD R, A

Load

Operation

R ← A

Description

The contents of the Accumulator are loaded into the Refresh Counter register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	A,I	X	2	ED, 4F

LD r, n

Load

Operation

r ← n

Description

The q operand is any of A, B, C, D, E, H, or L. The immediate n operand is loaded into the specified register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	r,n	X	2	kk, nn

kk = binary code **00 rrr 110** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

LD r, r’

Load

Operation

r ← r’

Description

The r or r’ operand is any of A, B, C, D, E, H, or L. The contents of the r’ register, which remain unchanged, are loaded into the r register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	r,r’	X	1	kk,

kk = binary code **01 ddd sss’** where **ddd** identifies the destination A, B, C, D, E, H, or L register and **sss** identifies the source A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	ddd or sss
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

LD r, (uu)

Load

Operation

r ← (uu)

Description

The uu operand is any of HL, IX+d, or IY+d and the r operand is any of A, B, C, D, E, H, or L. The specified r register is loaded from the memory location specified by the multi-byte uu register.

ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	r,(HL)	X	3	kk
LD.S	r,(HL)	1	4	52, kk
LD.L	r,(HL)	0	4	49, kk
LD	r,(IX+d)	X	4	DD, kk
LD.S	r,(IX+d)	1	5	52, DD, kk
LD.L	r,(IX+d)	0	5	49, DD, kk
LD	r,(IY+d)	X	4	FD, kk
LD.S	r,(IY+d)	1	5	52, FD, kk
LD.L	r,(IY+d)	0	5	49, FD, kk

kk = binary code **01 rrr 110** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

LD SP, rr

Load

Operation

SP ← rr

Description

The rr operand is any of HL, IX, or IY. The contents of the multi-byte rr register are loaded into the Stack Pointer.

In ADL mode or when the .L suffix is active, the destination is SPL. Otherwise, the destination is SPS. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	SP,HL	x	1	F9
LD.S	SP,HL	1	2	52, F9
LD.L	SP,HL	0	2	49, F9
LD	SP,IX	X	2	DD, F9
LD.S	SP,IX	1	3	52, DD, F9
LD.L	SP,IX	0	3	49, DD, F9
LD	SP,IY	X	2	FD, F9
LD.S	SP,IY	1	3	52, FD, F9
LD.L	SP,IY	0	3	49, FD, F9

LD tt,(HL)

Load

Operation

tt ← (HL)

Description

The tt operand is any of BC, DE, HL, IX, or IY. The contents of the memory location specified by the contents of the HL register are loaded into the multi-byte tt register.

ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	BC,(HL)	0/1	4/5	ED, 07
LD.S	BC,(HL)	1	5	52, ED, 07
LD.L	BC,(HL)	0	6	49, ED, 07
LD	DE,(HL)	0/1	4/5	ED, 17
LD.S	DE,(HL)	1	5	52, ED, 17
LD.L	DE,(HL)	0	6	49, ED, 17
LD	HL,(HL)	0/1	4/5	ED, 27
LD.S	HL,(HL)	1	5	52, ED, 27
LD.L	HL,(HL)	0	6	49, ED, 27
LD	IX,(HL)	0/1	4/5	ED, 37
LD.S	IX,(HL)	1	5	52, ED, 37
LD.L	IX,(HL)	0	6	49, ED, 37
LD	IY,(HL)	0/1	4/5	ED, 31
LD.S	IY,(HL)	1	5	52, ED, 31
LD.L	IY,(HL)	0	6	49, ED, 31

LD tt,(ii)

Load

Operation

tt ← (ii)

Description

The tt operand is any of BC, DE, HL, IX, or IY. The ii operand is any of IX+d or IY+d. The contents of the memory location specified by the contents of the ii register are loaded into the multi-byte tt register. ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	BC,(IX+d)	0/1	4/5	DD, 07, dd
LD.S	BC,(IX+d)	1	5	52, DD, 07, dd
LD.L	BC,(IX+d)	0	6	49, DD, 07, dd
LD	DE,(IX+d)	0/1	4/5	DD, 17, dd
LD.S	DE,(IX+d)	1	5	52, DD, 17, dd
LD.L	DE,(IX+d)	0	6	49, DD, 17, dd
LD	HL,(IX+d)	0/1	4/5	DD, 27, dd
LD.S	HL,(IX+d)	1	5	52, DD, 27, dd
LD.L	HL,(IX+d)	0	6	49, DD, 27, dd
LD	IX,(IX+d)	0/1	4/5	DD, 37, dd
LD.S	IX,(IX+d)	1	5	52, DD, 37, dd
LD.L	IX,(IX+d)	0	6	49, DD, 37, dd
LD	IY,(IX+d)	0/1	4/5	DD, 31, dd
LD.S	IY,(IX+d)	1	5	52, DD, 31, dd
LD.L	IY,(IX+d)	0	6	49, DD, 31, dd
LD	BC,(IY+d)	0/1	4/5	FD, 07, dd
LD.S	BC,(IY+d)	1	5	52, FD, 07, dd
LD.L	BC,(IY+d)	0	6	49, FD, 07, dd
LD	DE,(IY+d)	0/1	4/5	FD, 17, dd
LD.S	DE,(IY+d)	1	5	52, FD, 17, dd
LD.L	DE,(IY+d)	0	6	49, FD, 17, dd
LD	HL,(IY+d)	0/1	4/5	FD, 27, dd
LD.S	HL,(IY+d)	1	5	52, FD, 27, dd
LD.L	HL,(IY+d)	0	6	49, FD, 27, dd
LD	IX,(IY+d)	0/1	4/5	FD, 31, dd
LD.S	IX,(IY+d)	1	5	52, FD, 31, dd
LD.L	IX,(IY+d)	0	6	49, FD, 31, dd
LD	IY,(IY+d)	0/1	4/5	FD, 37, dd
LD.S	IY,(IY+d)	1	5	52, FD, 37, dd
LD.L	IY,(IY+d)	0	6	49, FD, 37, dd

LD (uu), n

Load

Operation

(uu) ← n

Description

The uu operand is any of HL, IX+d, or IY+d. The immediate n operand is loaded into the memory location specified by the multi-byte uu register.

ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(HL),n	X	3	36, nn
LD.S	(HL),n	1	4	52, 36, nn
LD.L	(HL),n	0	4	49, 36, nn
LD	(IX+d),n	X	4	DD, 36, nn
LD.S	(IX+d),n	1	5	52, DD, 36, nn
LD.L	(IX+d),n	0	5	49, DD, 36, nn
LD	(IY+d),n	X	4	FD, 36, nn
LD.S	(IY+d),n	1	5	52, FD, 36, nn
LD.L	(IY+d),n	0	5	49, FD, 36, nn

LD (uu), r

Load

Operation

(uu) ← r

Description

The uu operand is any of HL, IX+d, or IY+d and the r operand is any of A, B, C, D, E, H, or L. The specified R register is loaded into the memory location specified by the multi-byte uu register.

ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LD	(HL),r	X	3	kk
LD.S	(HL),r	1	4	52, kk
LD.L	(HL),r	0	4	49, kk
LD	(IX+d),r	X	4	DD, kk
LD.S	(IX+d),r	1	5	52, DD, kk
LD.L	(IX+d),r	0	5	49, DD, kk
LD	(IY+d),r	X	4	FD, kk
LD.S	(IY+d),r	1	5	52, FD, kk
LD.L	(IY+d),r	0	5	49, FD, kk

kk = binary code **01 110 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

LEA tt, IY+d

Load Effective Address

Operation

tt ← IY+d

Description

The tt operand is any of BC, DE, HL, IX, or IY. The contents of the IY register is added to the signed displacement d and the sum is loaded into the multi-byte tt register.

ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LEA	ss,IY+d	X	—	ED, kk, dd
LEA.S	ss,IY+d	1	—	52, ED, kk, dd
LEA.L	ss,IY+d	0	—	49, ED, kk, dd
LEA	IX,IY+d	X	—	ED, 54, dd
LEA.S	IX,IY+d	1	—	52, ED, 54, dd
LEA.L	IX,IY+d	0	—	49, ED, 54, dd

kk = binary code **00 ss0 011** where **ss** identifies the BC, DE, HL, or IY register assembled as follows in the object code:

Register	ss
BC	**00**
DE	**01**
HL	**10**
IY	**11**

LEA tt, IX+d

Load Effective Address

Operation

tt ← IX+d

Description

The tt operand is any of BC, DE, HL, IX, or IY. The contents of the IX register are added to the signed displacement d and the sum is loaded into the multi-byte tt register.

ADL mode affects operations with the BC, DE, HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
LEA	ss,IX+d	X	—	ED, kk, dd
LEA.S	ss,IX+d	1	—	52, ED, kk, dd
LEA.L	ss,IX+d	0	—	49, ED, kk, dd
LEA	IY,IX+d	X	—	ED, 55, dd
LEA.S	IY,IX+d	1	—	52, ED, 55, dd
LEA.L	IY,IX+d	0	—	49, ED, 55, dd

kk = binary code **00 ss0 010** where **ss** identifies the BC, DE, HL, or IX register assembled as follows in the object code:

Register	ss
BC	**00**
DE	**01**
HL	**10**
IX	**11**

MUL ss

Multiply

Operation

ss ← ss(low) x ss(high)

Description

The ss operand is any of BC, DE, HL, or SP. The register pair is replaced with the product of the register's low byte multiplied by its high byte. This is a 8- by 8-bit operation with a 16-bit result, regardless of the ADL mode.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
MUL	ss	X	6	ED, kk, dd
LD.S	ss,IY+d	1	—	52, ED, kk, dd
LD.L	ss,IY+d	0	—	49, ED, kk, dd
LD	IX,IY+d	X	—	ED, 54, dd
LD.S	IX,IY+d	1	—	52, ED, 54, dd
LD.L	IX,IY+d	0	—	49, ED, 54, dd

kk = binary code **01 ss1 100** where **ss** identifies the BC, DE, HL, or SP register assembled as follows in the object code:

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

OR A, s

Logical OR

Operation

A ← A ∨ s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is logically OR'ed to the contents of the Accumulator where the result is stored. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if carry from Bit 3; reset otherwise.
P/V Set if parity is even; reset otherwise.
N Reset.
C Reset.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OR	A,r	X	1	jj
OR	A,n	X	2	F6, nn
OR	A,(HL)	X	2	B6
OR.S	A,(HL)	1	3	52, B6
OR.L	A,(HL)	0	3	49, B6
OR	A,(IX+d)	X	4	DD, B6, dd
OR.S	A,(IX+d)	1	5	52, DD, B6, dd
OR.L	A,(IX+d)	0	5	49, DD, B6, dd
OR	A,(IY+d)	X	4	FD, B6, dd
OR.S	A,(IY+d)	1	5	52, FD, B6, dd
OR.L	A,(IY+d)	0	5	49, FD, B6, dd

jj = binary code **10 110 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

OTD2R

Output and Decrement HL - Repeat

Operation

repeat:
{ (BC) ← (HL)
B ← B – 1
C ← C – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The B, C, and HL registers are decremented. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTD2R	—	X	5/3	ED, BC
OTD2R.S	—	1	6/3	52, ED, BC
OTD2R.L	—	0	6/3	49, ED, BC

OTDM

Output and Decrement HL

Operation

(0,C) ← (HL)
B ← B – 1
C ← C – 1
HL ← HL – 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address 15-8 output to zeros. The B, C, and HL registers are decremented.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTDM	—	X	5	ED, 8B
OTDM.S	—	1	6	52, ED, 8B
OTDM.L	—	0	6	49, ED, 8B

OTDMR

Output and Decrement HL

Operation

repeat:
{ (0,C) ← (HL)
B ← B – 1
C ← C – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address (15-8) set to zero. The B, C, and HL registers are decremented. The instruction repeats until register B equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTDMR	—	X	5/3	ED, 9B
OTDMR.S	—	1	6/3	52, ED, 9B
OTDMR.L	—	0	6/3	49, ED, 9B

OTDR

Output and Decrement HL

Operation

repeat:
{ (0,C) ← (HL)
B ← B – 1
HL ← HL – 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address (15-8) set to zero. The B and HL registers are decremented. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTDR	—	X	5/3	ED, BB
OTDR.S	—	1	6/3	52, ED, BB
OTDR.L	—	0	6/3	49, ED, BB

OTI2R

Output and Increment HL - Repeat

Operation

repeat:
{ (BC) ← (HL)
B ← B – 1
C ← C + 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The C and HL registers are incremented and the B register decrements. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTI2R	—	X	5/3	ED, B4
OTI2R.S	—	1	6/3	52, ED, B4
OTI2R.L	—	0	6/3	49, ED, B4

OTIM

Output and Increment HL

Operation

(0,C) ← (HL)
B ← B – 1
C ← C + 1
HL ← HL + 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address (15-8) set to zero. The C and HL registers are incremented and the B register decrements.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTIM	—	X	5	ED, 83
OTIM.S	—	1	6	52, ED, 83
OTIM.L	—	0	6	49, ED, 83

OTIMR

Output and Increment HL

Operation

repeat:
{ (0,C) ← (HL)
B ← B – 1
C ← C + 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address (15-8) set to zero. The C and HL registers are incremented and register B decrements. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTIMR	—	X	5/3	ED, 93
OTIMR.S	—	1	6/3	52, ED, 93
OTIMR.L	—	0	6/3	49, ED, 93

OTIR

Output and Increment HL

Operation

repeat:
{ (0,C) ← (HL)
B ← B – 1
HL ← HL + 1
} while B ≠ 0

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the C register with address (15-8) set to zero. The B register decrements and the HL register increments. The instruction repeats until the B register equals zero.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OTIR	—	X	5/3	ED, B3
OTIR.S	—	1	6/3	52, ED, B3
OTIR.L	—	0	6/3	49, ED, B3

OUT (C), r

Output

Operation

(BC) ← r

Description

The r operand is any of the A, B, C, D, E, H, and L registers. The specified register is output to the I/O address specified by the BC register.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUT	(C),r	X	3	ED, kk

kk = binary code **01 rrr 001** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

OUTD

Output and Decrement HL

Operation

(BC) ← (HL)
B ← B – 1
HL ← HL – 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The B and HL registers are decremented.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUTD	—	X	5	ED, AB
OUTD.S	—	1	6	52, ED, AB
OUTD.L	—	0	6	49, ED, AB

OUTD2

Output and Decrement HL

Operation

(BC) ← (HL)
B ← B – 1
C ← C – 1
HL ← HL – 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The B, C, and HL registers are decremented.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUTD2	—	X	5	ED, AC
OUTD2.S	—	1	6	52, ED, AC
OUTD2.L	—	0	6	49, ED, AC

OUT (n), A

Output

Operation

(n) ← A

Description

The n operand is placed on address bus (7:0), the contents of the Accumulator on address bus (15:8), and zero on address bus (23:16). The contents of the Accumulator are output to this I/O address.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUT	(n),A	X	3	D3, nn

OUT0 (n), r

Output Register to I/O Port

Operation

(0,n) ← r

Description

The n operand is placed on address bus (7:0) and address bus (23:8) is zero. The byte at the specified register is output to this I/O address.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUT0	(n),r	X	4	ED, jj, nn

jj = binary code **00 rrr 001** where **rrr** identifies the A, B, C, D, E, H, or L registers assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

OUTI

Output and Increment HL

Operation

(BC) ← (HL)
B ← B – 1
HL ← HL + 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The B register decrements and the HL register increments.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUTI	—	X	5	ED, A3
OUTI.S	—	1	6	52, ED, A3
OUTI.L	—	0	6	49, ED, A3

OUTI2

Output and Increment HL

Operation

(BC) ← (HL)
B ← B – 1
C ← C + 1
HL ← HL + 1

Description

The contents of the memory location specified by the HL are loaded into the CPU. This byte is then output to the I/O address specified by the BC register. The B register decrements and the C and HL registers are incremented.

ADL mode affects operations with the HL register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Set if B = 0; reset otherwise.
H Not affected.
P/V Not affected.
N Set.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
OUTI2	—	X	5	ED, A4
OUTI2.S	—	1	6	52, ED, A4
OUTI2.L	—	0	6	49, ED, A4

PEA IX+d

Push Effective Address

Operation

if .L or (ADL and not .S)
SPL ← SPL – 3
(SPL) ← IX+d
else
SPS ← SPS – 2
(SPS) ← IX+d

Description

In ADL mode, SPL decrements by three and the 24-bit sum of the contents of IX and the signed displacement d is pushed onto the stack at SPL. In non-ADL mode, SPS decrements by two and the 16-bit sum of the contents of IX and the signed displacement d is pushed onto the stack at SPS.

ADL mode affects operations with the IX or SP register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
PEA	IX+d	0/1	5/6	ED, 65, dd
PEA.S	IX+d	1	6	52, ED, 65, dd
PEA.L	IX+d	0	7	49, ED, 65, dd

PEA IY+d

Push Effective Address

Operation

if .L or (ADL and not .S)
SPL ← SPL – 3
(SPL) ← IY+d
else
SPS ← SPS – 2
(SPS) ← IY+d

Description

In ADL mode, SPL decrements by three and the 24-bit sum of the contents of IY and the signed displacement d is pushed onto the stack at SPL. In non-ADL mode, SPS decrements by two and the 16-bit sum of the contents of IY and the signed displacement d is pushed onto the stack at SPS.

ADL mode affects operations with the IY or SP register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
PEA	IY+d	0/1	5/6	ED, 66, dd
PEA.S	IY+d	1	6	52, ED, 66, dd
PEA.L	IY+d	0	7	49, ED, 66, dd

POP vv

Pop Stack

Operation

if .L or (ADL and not .S)
vv7-0 ← (SPL)
vv15-8 ← (SPL+1)
vv23-16 ← (SPL+2)
SPL ← SPL + 3
else
vv7-0 ← (SPL)
vv15-8 ← (SPL+1)
SPS ← SPS + 2

Description

The vv operand is any of AF, BC, DE, HL, IX, or IY. In ADL mode, the three bytes at the memory location specified by the contents of SPL are loaded into the 24-bit vv register and the SPL increments by three. In non-ADL mode, the two bytes at the memory location specified by the contents of SPS are loaded into the 16-bit vv register and SPS increments by two. POP IX or POP IY in non-ADL mode results in the IX(23-16) or IY(23-16) being reset to zero. ADL mode affects operations with the AF, BC, DE, HL, IX, IY, or SP register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
POP	ss	0/1	3/4	kk
POP.S	ss	1	4	52, kk
POP.L	ss	0	5	49, kk
POP	IX	0/1	4/5	DD, E1
POP.S	IX	1	5	52, DD, E1
POP.L	IX	0	6	49, DD, E1
POP	IY	0/1	4/5	FD, E1
POP.S	IY	1	5	52, FD, E1
POP.L	IY	0	6	49, FD, E1

kk = binary code **11 ss0 001** where **ss** identifies the AF, BC, DE, or HL register assembled as follows into the object code:

Register	ss
AF	**11**
BC	**00**
DE	**01**
HL	**10**

PUSH vv

Push Stack

Operation

```
if .L or (ADL and not .S)
        (SPL–1) ← vv23-16
        (SPL–2) ← vv15-8
        (SPL–3) ← vv7-0
        SPL ← SPL – 3
else
        (SPL–1) ← vv15-86
        (SPL–2) ← vv7-08
        SPS ← SPS – 2
```

Description

The vv operand is any of AF, BC, DE, HL, IX, or IY. In ADL mode, the 24-bit vv register is loaded into the three bytes at the memory location specified by the contents of SPL-3 and the SPL decrements by three. In non-ADL mode, the 16-bit vv register is loaded into the two bytes at the memory location specified by the contents of SPS-2 and the SPS increments by two. ADL mode affects operations with the AF, BC, DE, HL, IX, IY, and SP registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
PUSH	ss	0/1	3/4	kk
PUSH.S	ss	1	4	52, kk
PUSH.L	ss	0	5	49, kk
PUSH	IX	0/1	4/5	DD, E5
PUSH.S	IX	1	5	52, DD, E5
PUSH.L	IX	0	6	49, DD, E5
PUSH	IY	0/1	4/5	FD, E5
PUSH.S	IY	1	5	52, FD, E5
PUSH.L	IY	0	6	49, FD, E5

kk = binary code **11 ss0 101** where **ss** identifies the AF, BC, DE, or HL register assembled as follows into the object code:

Register	ss
AF	**11**
BC	**00**
DE	**01**
HL	**10**

RES b, m

Reset Bit

Operation

mb ← 0

Description

The m operand is any of r, (HL), (IX+d), or (IY+d). This instruction resets Bit b in the specified register or memory location. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RES	b,r	X	2	CB, jj
RES	b,(HL)	X	3	CB, kk
RES.S	b,(HL)	1	4	52, CB, kk
RES.L	b,(HL)	0	4	49, CB, kk
RES	b,(IX+d)	X	5	DD, CB, dd, kk
RES.S	b,(IX+d)	1	6	52, DD, CB, dd, kk
RES.L	b,(IX+d)	0	6	49, DD, CB, dd, kk
RES	b,(IY+d)	X	5	FD, CB, dd, kk
RES.S	b,(IY+d)	1	6	52, FD, CB, dd, kk
RES.L	b,(IY+d)	0	6	49, FD, CB, dd, kk

jj = binary code **10 bbb rrr**, and **kk** = binary code **10 bbb 110**; where **rrr** identifies the A, B, C, D, E, H, or L register and **bbb** identifies the bit tested assembled as follows into the object code:

Bit Tested	bbb	Register	rrr
0	**000**	A	**111**
1	**001**	B	**000**
2	**010**	C	**001**
3	**011**	D	**010**
4	**100**	E	**011**
5	**101**	H	**100**
6	**110**	L	**101**
7	**111**		

RET

Return from Subroutine

Operation

PC ← (SP)

Description

ADL	Prefix	Operation
0	None	Pop 2-byte logical return address using SPS mapped by MBASE into PC. Keep ADL cleared to 0.
1	None	Pop 3-byte return address using SPL into PC. Keep ADL set to 1.
0	.S or .L	Pop a byte using SPL and load its units bit into ADL. — If ADL is still cleared to 0, pop a 2-byte logical return address using SPS mapped by MBASE into PC. —If ADL is now set to 1, pop a byte from SPL into PC (23-16). Then pop two bytes from SPS mapped by MBASE into PC (15-0).
1	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is now cleared to 0, pop a 2-byte logical address from SPL into PC. —If ADL is still set to 1, pop a 3-byte address from SPL into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RET	—	0/1	*	C9
RET.S	—	0	*	40, C9
RET.S	—	1	*	52, C9
RET.L	—	0	*	49, C9
RET.L	—	1	*	5B, C9

RET cc

Conditional Return from Subroutine

Operation

If cc true
PC ← (SP)

Description

ADL	Prefix	Operation
0	None	Pop a 2-byte logical return address using SPS mapped by MBASE into PC. Keep ADL cleared to 0.
1	None	Pop a 3-byte return address using SPL into PC. Keep ADL set to 1.
0	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is still cleared to 0, pop a 2-byte logical return address using SPS mapped by MBASE into PC. —If ADL is now set to 1, pop a byte from SPL into PC (23-16). Then pop two bytes from SPS mapped by MBASE into PC (15-0).
1	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is now cleared to 0, pop a 2-byte logical address from SPL into PC. —If ADL is still set to 1, pop a 3-byte address from SPL into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RET	cc	0/1	*	kk
RET.S	cc	0	*	40, kk
RET.S	cc	1	*	52, kk
RET.L	cc	0	*	49, kk
RET.L	cc	1	*	5B, kk

kk = binary code **11 ccc 000** where **ccc** identifies the condition for test assembled as follows into the object code:

Condition	ccc
NZ	**000**
Z	**001**
NC	**010**
C	**011**
PO	**100**
PE	**101**
P	**110**
M	**111**

RETI

Return from Interrupt

Operation

PC ← (SP)

Description

ADL	Prefix	Operation
0	None	Pop 2-byte logical return address using SPS mapped by MBASE into PC. Keep ADL cleared to 0.
1	None	Pop 3-byte return address using SPL into PC. Keep ADL set to 1.
0	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is still cleared to 0, pop a 2-byte logical return address using SPS mapped by MBASE into PC. —If ADL is now set to 1, pop a byte from SPL into PC (23-16). Then pop two bytes from SPS mapped by MBASE into PC (15-0).
1	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is now cleared to 0, pop a 2-byte logical address from SPL into PC. —If ADL is still set to 1, pop a 3-byte address from SPL into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RETI	—	0/1	*	ED, 4D
RETI.S	—	0	*	40, ED, 4D
RETI.S	—	1	*	52, ED, 4D
RETI.L	—	0	*	49, ED, 4D
RETI.L	—	1	*	5B, ED, 4D

RETN

Return from Nonmaskable Interrupt

Operation

PC ← (SP)

Description

ADL	Prefix	Operation
0	None	Pop a 2-byte logical return address using SPS mapped by MBASE into PC. Keep ADL cleared to 0.
1	None	Pop a 3-byte return address using SPL into PC. Keep ADL set to 1.
0	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is still cleared to 0, pop a 2-byte logical return address using SPS mapped by MBASE into PC. —If ADL is now set to 1, pop a byte from SPL into PC (23-16). Then pop two bytes from SPS mapped by MBASE into PC (15-0).
1	.S or .L	Pop a byte using SPL and load its units bit into ADL. —If ADL is now cleared to 0, pop a 2-byte logical address from SPL into PC. —If ADL is still set to 1, pop a 3-byte address from SPL into PC.

Condition Bits Affected

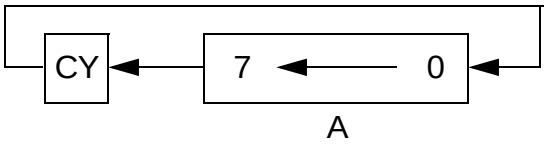
None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RETI	—	0/1	*	ED, 45
RETI.S	—	0	*	40, ED, 45
RETI.S	—	1	*	52, ED, 45
RETI.L	—	0	*	49, ED, 45
RETI.L	—	1	*	5B, ED, 45

RLA

Rotate Left Accumulator

Operation



Description

The contents of the Accumulator are rotated left one bit position. Bit 7 is copied into the Carry Flag and the previous contents of the Carry Flag are copied into Bit 0 of the m operand.

Condition Bits Affected

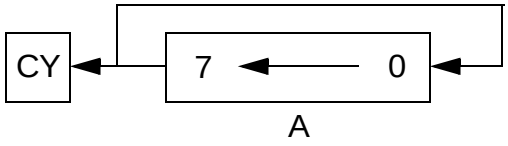
S Not affected.
Z Not affected.
H Reset.
P/V Not affected.
N Reset.
C Data from Bit 7 of the Accumulator.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RLA	—	X	1	17

RLCA

Rotate Left Carry Accumulator

Operation



Description

The contents of the Accumulator are rotated left one bit position. Bit 7 is copied into the Carry Flag and into Bit 0.

Condition Bits Affected

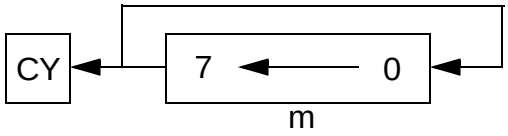
S Not affected.
Z Not affected.
H Reset.
P/V Not affected.
N Reset.
C Data from Bit 7 of the Accumulator

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RLCA	—	X	1	07

RLC m

Rotate Left Carry

Operation



Description

The m operand is any of A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d). The contents of the m operand are rotated left one bit position. Bit 7 is copied into the Carry Flag and into Bit 0 of the m operand. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 7 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RLC	r	X	2	CB, kk
RLC	(HL)	X	5	CB, 06
RLC.S	(HL)	1	6	52, CB, 06
RLC.L	(HL)	0	6	49, CB, 06
RLC	(IX+d)	X	7	DD, CB, dd, 06
RLC.S	(IX+d)	1	8	52, DD, CB, dd, 06
RLC.L	(IX+d)	0	8	49, DD, CB, dd, 06
RLC	(IY+d)	X	7	FD, CB, dd, 06
RLC.S	(IY+d)	1	8	52, FD, CB, dd, 06
RLC.L	(IY+d)	0	8	49, FD, CB, dd, 06

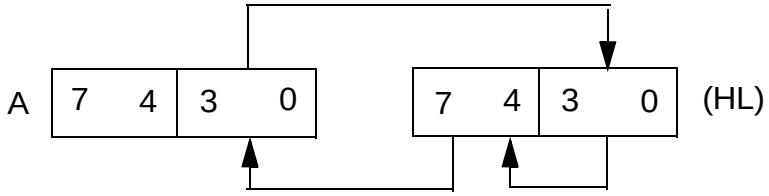
kk = binary code **00 000 rrr** where **rrr** identifies the A, B, C, D, E, H, and L registers assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

RLD

Rotate Left Digit

Operation



Description

The contents of the low-order four bits of the memory location (HL) are copied into the high-order four bits of the same memory location. The previous contents of the high-order four bits of the memory location are copied into the low-order four bits of the Accumulator. The previous contents of the low-order four bits of the Accumulator are copied into the low-order four bits of the memory location.

Condition Bits Affected

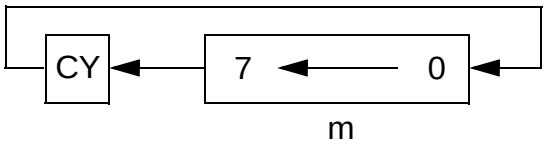
S Set if the Accumulator is negative; reset otherwise.
Z Set if the Accumulator is zero; reset otherwise.
H Reset.
P/V Set if parity of the Accumulator is even; reset otherwise.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RLD	—	X	5	ED, 6F

RL m

Rotate Left

Operation



Description

The m operand is any of A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d). The contents of the m operand are rotated left one bit position. Bit 7 is copied into the Carry Flag and the previous contents of the Carry Flag are copied into Bit 0 of the m operand. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 7 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RL	r	X	2	CB, kk
RL	(HL)	X	5	CB, 16
RL.S	(HL)	1	6	52, CB, 16
RL.L	(HL)	0	6	49, CB, 16
RL	(IX+d)	X	7	DD, CB, dd, 16
RL.S	(IX+d)	1	8	52, DD, CB, dd, 16
RL.L	(IX+d)	0	8	49, DD, CB, dd, 16
RL	(IY+d)	X	7	FD, CB, dd, 16
RL.S	(IY+d)	1	8	52, FD, CB, dd, 16
RL.L	(IY+d)	0	8	49, FD, CB, dd, 16

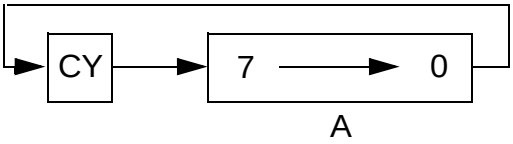
kk = binary code **00 010 rrr** where **rrr** identifies the registers A, B, C, D, E, H, or L assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

RRA

Rotate Left Accumulator

Operation



Description

The contents of the Accumulator are rotated right one bit position. The contents of Bit 0 are copied into the Carry Flag and the previous contents of the Carry Flag are copied into Bit 7.

Condition Bits Affected

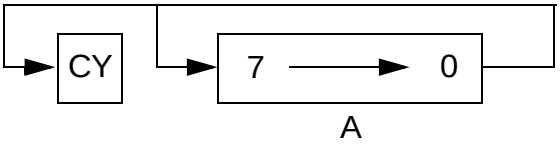
S Not affected.
Z Not affected.
H Reset.
P/V Not affected.
N Reset.
C Data from Bit 0 of the Accumulator.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RRA	—	X	1	1F

RRCA

Rotate Left Carry Accumulator

Operation



Description

The contents of the Accumulator are rotated right one bit position. The contents of Bit 0 are copied into the Carry Flag and into Bit 7.

Condition Bits Affected

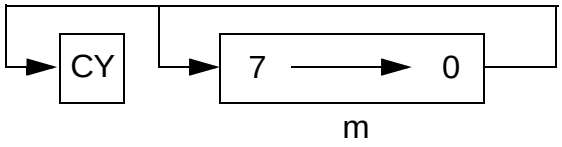
S Not affected.
Z Not affected.
H Reset.
P/V Not affected.
N Reset.
C Data from Bit 0 of the Accumulator.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RRCA	—	X	1	0F

RRC m

Rotate Left Carry

Operation



Description

The m operand is any of A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d). The contents of the m operand are rotated right one bit position. The contents of Bit 0 are copied into the Carry Flag and into Bit 7 of the m operand. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 0 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RRC	r	X	2	CB, kk
RRC	(HL)	X	5	CB, 0E
RRC.S	(HL)	1	6	52, CB, 0E
RRC.L	(HL)	0	6	49, CB, 0E
RRC	(IX+d)	X	7	DD, CB, dd, 0E
RRC.S	(IX+d)	1	8	52, DD, CB, dd, 0E
RRC.L	(IX+d)	0	8	49, DD, CB, dd, 0E
RRC	(IY+d)	X	7	FD, CB, dd, 0E
RRC.S	(IY+d)	1	8	52, FD, CB, dd, 0E
RRC.L	(IY+d)	0	8	49, FD, CB, dd, 0E

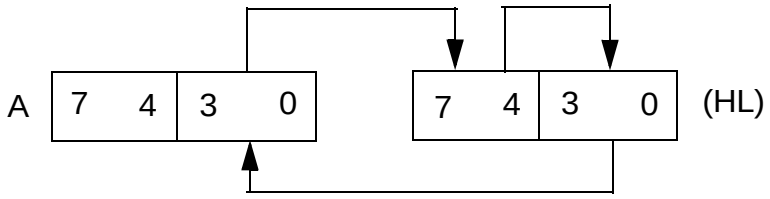
kk = binary code **00 001 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

RRD

Rotate Right Digit

Operation



Description

The contents of the low-order four bits of the memory location (HL) are copied into the low-order four bits of the Accumulator. The previous contents of the low-order four bits of the Accumulator are copied into high-order four bits of the same memory location. The previous contents of the high-order four bits of the memory location are copied into the low-order four bits of the memory location.

Condition Bits Affected

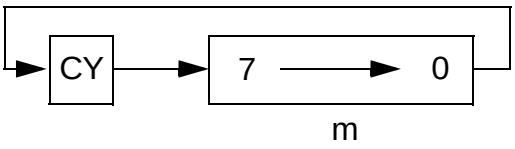
S Set if the Accumulator is negative; reset otherwise.
Z Set if the Accumulator is zero; reset otherwise.
H Reset.
P/V Set if parity of the Accumulator is even; reset otherwise.
N Reset.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RRD	—	X	5	ED, 67

RR m

Rotate Left

Operation



Description

The m operand is any of A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d). The contents of the m operand are rotated right one bit position. The contents of Bit 0 are copied into the Carry Flag and the previous contents of the Carry Flag are copied into Bit 7 of the m operand. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 0 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RR	r	X	2	CB, kk
RR	(HL)	X	5	CB, 1E
RR.S	(HL)	1	6	52, CB, 1E
RR.L	(HL)	0	6	49, CB, 1E
RR	(IX+d)	X	7	DD, CB, dd, 1E
RR.S	(IX+d)	1	8	52, DD, CB, dd, 1E
RR.L	(IX+d)	0	8	49, DD, CB, dd, 1E
RR	(IY+d)	X	7	FD, CB, dd, 1E
RR.S	(IY+d)	1	8	52, FD, CB, dd, 1E
RR.L	(IY+d)	0	8	49, FD, CB, dd, 1E

kk = binary code **00 011 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

RSMIX

Reset Mixed Mode

Operation

MADL ← 0

Description

The Mixed ADL Mode Flag is reset.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RSMIX	—	X	2	ED, 7E

RST n

Restart

Operation

—

Description

ADL	Prefix	Operation
0	None	Stack 2-byte logical return address using SPS mapped by MBASE. Keep ADL cleared to 0. Load the 16-bit logical address 00nn into PC.
1	None	Stack 3-byte return address using SPL. Keep ADL set to 1. Load the 24-bit address 0000nn into PC.
0	.IS	Stack 2-byte logical return address using SPS mapped by MBASE. Stack a **00** byte using SPL. Keep ADL cleared to 0. Load the 16-bit logical address **00nn** into PC.
1	.IS	Stack the two LS bytes of return address using SPS mapped by MBASE . Stack an MS byte of return address using SPL. Stack a **01** byte using SPL. Clear ADL to 0. Load the 16-bit logical address **00nn** into PC.
0	.IL	Stack 2-byte logical return address using SPL. Stack a **01** byte using SPL. Set ADL to 1. Load the 24-bit address **0000nn** into PC.
1	.IL	Stack 3-byte return address using SPL and keep ADL set to 1. Stack a **01** byte using SPL. Load the 24-bit address **0000nn** into PC.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RST	p	0/1	*	kk
RST.IS	p	0	*	40, C9
RST.IL	p	1	*	52, C9

kk = binary code **11 ttt 111** where **ttt** identifies the low-order byte to be loaded into the PC assembled as follows in the object code:

p	ttt
00H	**000**
08H	**001**
10H	**010**
18H	**011**
20H	**100**
28H	**101**
30H	**110**
38H	**111**

SBC A, s

Subtract with Carry

Operation

A ← A – s – CY

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand, along with the Carry Flag (c in the F register) is subtracted from the contents of the Accumulator, which contains the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if overflow; reset otherwise.
N Set.
C Set if borrow; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SBC	A,r	X	1	jj
SBC	A,n	X	2	DE, nn
SBC	A,(HL)	X	2	9E
SBC.S	A,(HL)	1	3	52, 9E
SBC.L	A,(HL)	0	3	49, 9E
SBC	A,(IX+d)	X	4	DD, 9E, dd
SBC.S	A,(IX+d)	1	5	52, DD, 9E, dd
SBC.L	A,(IX+d)	0	5	49, DD, 9E, dd
SBC	A,(IY+d)	X	4	FD, 9E, dd
SBC.S	A,(IY+d)	1	5	52, FD, 9E, dd
SBC.L	A,(IY+d)	0	5	49, FD, 9E, dd

jj = binary code **10 011 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

SBC HL, ss

Subtract with Carry

Operation

HL ← HL – ss – CY

Description

The ss operand is any of the BC, DE, HL, or SP registers. The ss operand, along with the Carry Flag (c in the F register) is subtracted from the contents of the HL register, which contains the result.

ADL mode affects operations with the HL, BC, DE, and SP registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if borrow from Bit 12; reset otherwise.
P/V Set if overflow; reset otherwise.
N Set.
C Set if borrow; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SBC	HL,ss	X	2	ED, kk
SBC.S	HL,ss	1	3	52, ED, kk
SBC.L	HL,ss	0	3	49, ED, kk

kk = binary code **01 ss0 010** where **ss** identifies the BC, DE, HL, or SP register assembled as follows into the object code:

Register	ss
BC	**00**
DE	**01**
HL	**10**
SP	**11**

SCF

Set Carry Flag

Operation

CY ← 1

Description

The ss operand is any of the BC, DE, HL, or SP registers. The ss operand, along with the Carry Flag (c in the F register) is subtracted from the contents of the HL register, which contains the result.

ADL mode affects operations with the HL, BC, DE, and SP registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Not affected.
H Reset.
P/V Not affected.
N Reset.
C Set.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SCF	—	X	1	37

SET b, m

Set BIT

Operation

mb ← 1

Description

The m operand is any of r, (HL), (IX+d), or (IY+d). Bit b in the specified register or memory location is set. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Not affected.
Z Not affected.
H Not affected.
P/V Not affected.
N Not affected.
C Not affected.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SET	b,r	X	2	CB, jj
SET	b,(HL)	X	3	CB, kk
SET.S	b,(HL)	1	4	52, CB, kk
SET.L	b,(HL)	0	4	49, CB, kk
SET	b,(IX+d)	X	5	DD, CB, dd, kk
SET.S	b,(IX+d)	1	6	52, DD, CB, dd, kk
SET.L	b,(IX+d)	0	6	49, DD, CB, dd, kk
SET	b,(IY+d)	X	5	FD, CB, dd, kk
SET.S	b,(IY+d)	1	6	52, FD, CB, dd, kk
SET.L	b,(IY+d)	0	6	49, FD, CB, dd, kk

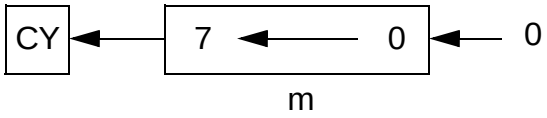
jj = binary code **11 bbb rrr** and **kk** = binary code **11 bbb 110**; where **rrr** identifies the A, B, C, D, E, H, or L register and **bbb** identifies the bit tested assembled as follows into the object code:

BIT Tested	bbb	Register	rrr
0	**000**	A	**111**
1	**001**	B	**000**
2	**010**	C	**001**
3	**011**	D	**010**
4	**100**	E	**011**
5	**101**	H	**100**
6	**110**	L	**101**
7	**111**		

SLA m

Shift Left Arithmetic

Operation



Description

The m operand is any of A, B, C, D, E, H, L, (HL), (IX+d), or (IY+d). The contents of the m operand are shifted left one bit position. Bit 7 is copied into the Carry Flag and a zero is copied into Bit 0 of the m operand. ADL mode affects operations with register HL, IX, or IY. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 7 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SLA	r	X	2	CB, kk
SLA	(HL)	X	5	CB, 16
SLA.S	(HL)	1	6	52, CB, 16
SLA.L	(HL)	0	6	49, CB, 16
SLA	(IX+d)	X	7	DD, CB, dd, 16
SLA.S	(IX+d)	1	8	52, DD, CB, dd, 16
SLA.L	(IX+d)	0	8	49, DD, CB, dd, 16
SLA	(IY+d)	X	7	FD, CB, dd, 16
SLA.S	(IY+d)	1	8	52, FD, CB, dd, 16
SLA.L	(IY+d)	0	8	49, FD, CB, dd, 16

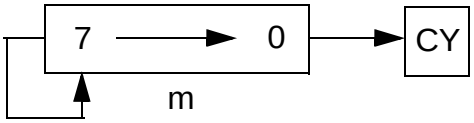
kk = binary code **00 100 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

SRA m

Shift Right Arithmetic

Operation



Description

The contents of the m operand are shifted right one bit position. The contents of Bit 0 are copied into the Carry Flag and the previous contents of Bit 7 are unchanged. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 0 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SRA	r	X	2	CB, kk
SRA	(HL)	X	5	CB, 2E
SRA.S	(HL)	1	6	52, CB, 2E
SRA.L	(HL)	0	6	49, CB, 2E
SRA	(IX+d)	X	7	DD, CB, dd, 2E
SRA.S	(IX+d)	1	8	52, DD, CB, dd, 2E
SRA.L	(IX+d)	0	8	49, DD, CB, dd, 2E
SRA	(IY+d)	X	7	FD, CB, dd, 2E
SRA.S	(IY+d)	1	8	52, FD, CB, dd, 2E
SRA.L	(IY+d)	0	8	49, FD, CB, dd, 2E

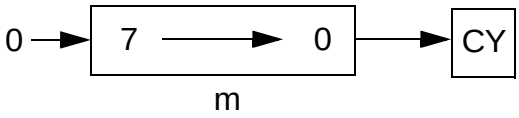
kk = binary code **00 101 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

SRL m

Shift Right Logical

Operation



Description

The contents of the m operand are shifted right one bit position. The contents of Bit 0 are copied into the Carry Flag and Bit 7 is reset. ADL mode affects operations with the HL, IX, or IY register. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Reset.
P/V Set if parity is even; reset otherwise.
N Reset.
C Data from Bit 0 of the source.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SRL	r	X	2	CB, kk
SRL	(HL)	X	5	CB, 3E
SRL.S	(HL)	1	6	52, CB, 3E
SRL.L	(HL)	0	6	49, CB, 3E
SRL	(IX+d)	X	7	DD, CB, dd, 3E
SRL.S	(IX+d)	1	8	52, DD, CB, dd, 3E
SRL.L	(IX+d)	0	8	49, DD, CB, dd, 3E
SRL	(IY+d)	X	7	FD, CB, dd, 3E
SRL.S	(IY+d)	1	8	52, FD, CB, dd, 3E
SRL.L	(IY+d)	0	8	49, FD, CB, dd, 3E

kk = binary code **00 111 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows in the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

STMIX
Reset Mixed Mode

Operation

MADL ← 1

Description

The Mixed ADL Mode Flag is set to 1.

Condition Bits Affected

None

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
RSMIX	—	X	2	ED, 7D

SUB A, s

Subtract

Operation

A ← A – s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is subtracted from the contents of the Accumulator, which contains the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set if borrow from Bit 4; reset otherwise.
P/V Set if overflow; reset otherwise.
N Set.
C Set if borrow; reset otherwise.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
SUB	A,r	X	1	jj
SUB	A,n	X	2	D6, nn
SUB	A,(HL)	X	2	96
SUB.S	A,(HL)	1	3	52, 96
SUB.L	A,(HL)	0	3	49, 96
SUB	A,(IX+d)	X	4	DD, 96, dd
SUB.S	A,(IX+d)	1	5	52, DD, 96, dd
SUB.L	A,(IX+d)	0	5	49, DD, 96, dd
SUB	A,(IY+d)	X	4	FD, 96, dd
SUB.S	A,(IY+d)	1	5	52, FD, 96, dd
SUB.L	A,(IY+d)	0	5	49, FD, 96, dd

jj = binary code **10 010 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

TST A, p

Test

Operation

A • p

Description

The p operand is any of r, n, or (HL). The p operand is bitwise AND'ed with the contents of the Accumulator and the flags are set to 1. The contents of the Accumulator and the p operand are not altered.

ADL mode affects operations with registers HL. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set.
P/V Set if parity is even; reset otherwise.
N Reset.
C Reset.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
TST	A,r	X	2	ED, jj
TST	A,n	X	3	ED, 64, nn
TST	A,(HL)	X	3	ED, 74
TST.S	A,(HL)	1	4	52, ED, 74
TST.L	A,(HL)	0	4	49, ED, 74

jj = binary code **00 rrr 100** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

TSTIO n

Test I/O byte

Operation

(0,C) • n

Description

The contents of the C register are placed on the address bus with address (15-8) as zeros. The I/O at this address is bitwise AND'ed with the immediate n value and the flags are set to 1.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set.
P/V Set if parity is even; reset otherwise.
N Reset.
C Reset.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
TSTIO	n	X	4	ED, 74, nn

XOR A, s

Logical Exclusive-OR

Operation

A ← A ⊕ s

Description

The s operand is any of r, n, (HL), (IX+d), or (IY+d). The s operand is bitwise exclusive-OR'ed with the contents of the Accumulator, which contains the result. ADL mode affects operations with the HL, IX, and IY registers. ADL mode may be overridden with the .S or .L suffix.

Condition Bits Affected

S Set if result is negative; reset otherwise.
Z Set if result is zero; reset otherwise.
H Set.
P/V Set if parity is even; reset otherwise.
N Reset.
C Reset.

Mnemonic	Operands	ADL Mode	Cycles	Op Codes
XOR	A,r	X	1	jj
XOR	A,n	X	2	EE, nn
XOR	A,(HL)	X	2	AE
XOR.S	A,(HL)	1	3	52, AE
XOR.L	A,(HL)	0	3	49, AE
XOR	A,(IX+d)	X	4	DD, AE, dd
XOR.S	A,(IX+d)	1	5	52, DD, AE, dd
XOR.L	A,(IX+d)	0	5	49, DD, AE, dd
XOR	A,(IY+d)	X	4	FD, AE, dd
XOR.S	A,(IY+d)	1	5	52, FD, AE, dd
XOR.L	A,(IY+d)	0	5	49, FD, AE, dd

jj = binary code **10 101 rrr** where **rrr** identifies the A, B, C, D, E, H, or L register assembled as follows into the object code:

Register	rrr
A	**111**
B	**000**
C	**001**
D	**010**
E	**011**
H	**100**
L	**101**

Op Code Maps

Table 7. Op Code Map (First Op Code)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ d	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,d	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD (HL),nn	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR C,d	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	.SIS prefix	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD C,A	LD C,B	.LIS prefix	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	.SIL prefix	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	.LIL prefix	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	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
7	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
8	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
9	SUB A,B	SUB A,C	SUB A,D	SUB A,E	SUB A,H	SUB A,L	SUB A,(HL)	SUB A,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
A	AND A,B	AND A,C	AND A,D	AND A,E	AND A,H	AND A,L	AND A,(HL)	AND A,A	XOR A,B	XOR A,C	XOR A,D	XOR A,E	XOR A,H	XOR A,L	XOR A,(HL)	XOR A,A
B	OR A,B	OR A,C	OR A,D	OR A,E	OR A,H	OR A,L	OR A,(HL)	OR A,A	CP A,B	CP A,C	CP A,D	CP A,E	CP A,H	CP A,L	CP A,(HL)	CP A,A
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST 0	RET Z	RET	JP Z,nn	Table 8	CALL Z,nn	CALL nn	ADC A,n	RST 8
D	RET NZ	POP DE	JP NC,nn	OUT (n),A	CALL NC,nn	PUSH DE	SUB A,n	RST 10H	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn	Table 9	SBC A,n	RST 18H
E	RET PO	POP HL	JP PO,nn	EX (SP),HL	CALL PO,nn	PUSH HL	AND A,n	RST 20	RET PE	JP (HL)	JP PE,nn	EX DE,HL	CALL PE,nn	Table 10	XOR A,n	RST 28H
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR A,n	RST 30H	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn	Table 11	CP A,n	RST 38H

Note: n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

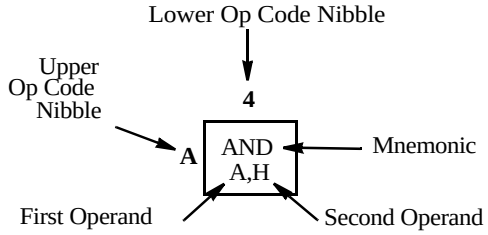


Table 8. Op Code Map (Second Op Code after 0CBH)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	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
1	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
2	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
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	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
5	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
6	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
7	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
8	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
9	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
A	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
B	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
C	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 1,L	SET 1,(HL)	SET 1,A
D	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
E	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
F	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

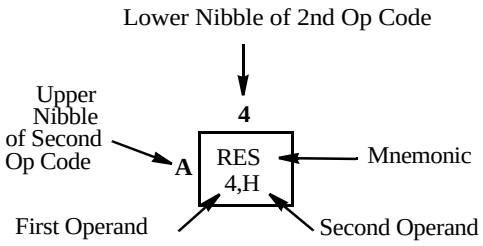


Table 9. Op Code Map (Second Op Code After 0DDH)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0								LD BC, (IX±d)		ADD IX,BC						LD (IX ±d),BC
1								LD DE, (IX±d)		ADD IX,DE						LD (IX ±d),DE
2		LD IX,nn	LD (nn),IX	INC IX				LD HL, (IX±d)		ADD IX,IX	LD IX,(nn)	DEC IX				LD (IX ±d),HL
3		LD IY, (IX±d)			INC (IX±d)	DEC (IX±d)	LD (IX ±d),n	LD IX, (IX±d)		ADD IX,SP					LD (IX ±d),IY	LD (IX ±d),IX
4							LD B, (IX±d)								LD C, (IX±d)	
5							LD D, (IX±d)								LD E, (IX±d)	
6							LD H, (IX±d)								LD L, (IX±d)	
7	LD (IX ±d),B	LD (IX ±d),C	LD (IX ±d),D	LD (IX ±d),E	LD (IX ±d),H	LD (IX ±d),L		LD (IX ±d),A							LD A, (IX±d)	
8							ADD A, (IX±d)								ADC A, (IX±d)	
9							SUB A, (IX±d)								SBC A, (IX±d)	
A							AND A, (IX±d)								XOR A, (IX±d)	
B							OR A, (IX±d)								CP A, (IX±d)	
C												Table 12				
D																
E		POP IX		EX (SP),IX		PUSH IX				JP (IX)						
F										LD SP,IX						

Note: n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

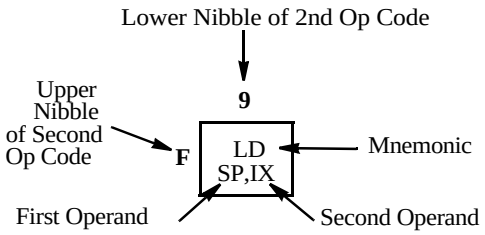


Table 10. Op Code Map (Second Op Code After 0EDH)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	IN0 B,(n)	OUT0 (n),B	LEA BC ,IX±d	LEA BC ,IY±d	TST A,B			LD BC, (HL)	IN0 C,(n)	OUT0 (n),C			TST A,C			LD (HL) ,BC
1	IN0 D,(n)	OUT0 (n),D	LEA DE ,IX±d	LEA DE ,IY±d	TST A,D			LD DE, (HL)	IN0 E,(n)	OUT0 (n),E			TST A,E			LD (HL) ,DE
2	IN0 H,(n)	OUT0 (n),H	LEA HL ,IX±d	LEA HL ,IY±d	TST A,H			LD HL, (HL)	IN0 L,(n)	OUT0 (n),L			TST A,L			LD (HL) ,HL
3		LD IY, (HL)	LEA IX ,IX±d	LEA IY ,IY±d	TST A,(HL)			LD IX, (HL)	IN0 A,(n)	OUT0 (n),A			TST A,A		LD (HL) ,IY	LD (HL) ,IX
4	IN B,(C)	OUT (C),B	SBC HL,BC	LD (nn),BC	NEG	RETN	IM 0	LD I,A	IN C,(C)	OUT (C),C	ADC HL,BC	LD BC,(nn)	MLT BC	RETI		LD R,A
5	IN D,(C)	OUT (C),D	SBC HL,DE	LD (nn),DE	LEA IX ,IY±d	LEA IY ,IX±d	IM 1	LD A,I	IN E,(C)	OUT (C),E	ADC HL,DE	LD DE,(nn)	MLT DE		IM 2	LD A,R
6	IN H,(C)	OUT (C),H	SBC HL,HL	LD (nn),HL	TST A,n	PEA IX±d	PEA IY±d	RRD	IN L,(C)	OUT (C),L	ADC HL,HL	LD HL,(nn)	MLT HL	LD MB,A	LD A,MB	RLD
7			SBC HL,SP	LD (nn),SP	TSTIO n		SLP		IN A,(C)	OUT (C),A	ADC HL,SP	LD SP,(nn)	MLT SP	STMIX	RSMIX	
8			INIM	OTIM	INI2						INDM	OTDM	IND2			
9			INIMR	OTIMR	INI2R						INDMR	OTDMR	IND2R			
A	LDI	CPI	INI	OUTI	OUTI2				LDD	CPD	IND	OUTD	OUTD2			
B	LDIR	CPIR	INIR	OTIR	OTI2R				LDDR	CPDR	INDR	OTDR	OTD2R			
C																
D																
E																
F																

Note: n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

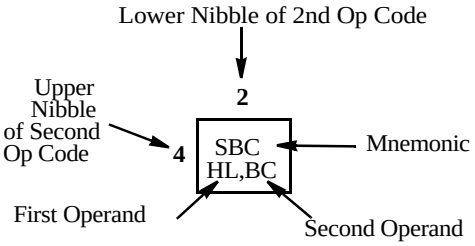


Table 11. Op Code Map (Second Op Code After 0FDH)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0								LD BC, (IY±d)		ADD IY,BC						LD (IY ±d),BC
1								LD DE, (IY±d)		ADD IY,DE						LD (IY ±d),DE
2		LD IY,nn	LD (nn),IY	INC IY				LD HL, (IY±d)		ADD IY,IY	LD IY,(nn)	DEC IY				LD (IY ±d),HL
3		LD IX, (IY±d)			INC (IY±d)	DEC (IY±d)	LD (IY ±d),n	LD IY, (IY±d)		ADD IY,SP					LD (IY ±d),IX	LD (IY ±d),IY
4							LD B, (IY±d)								LD C, (IY±d)	
5							LD D, (IY±d)								LD E, (IY±d)	
6							LD H, (IY±d)								LD L, (IY±d)	
7	LD (IY ±d),B	LD (IY ±d),C	LD (IY ±d),D	LD (IY ±d),E	LD (IY ±d),H	LD (IY ±d),L		LD (IY ±d),A							LD A, (IY±d)	
8							ADD A, (IY±d)								ADC A, (IY±d)	
9							SUB A, (IY±d)								SBC A, (IY±d)	
A							AND A, (IY±d)								XOR A, (IY±d)	
B							OR A, (IY±d)								CP A, (IY±d)	
C												Table 13				
D																
E		POP IY		EX (SP),IY		PUSH IY				JP (IY)						
F										LD SP,IY						

Note: n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

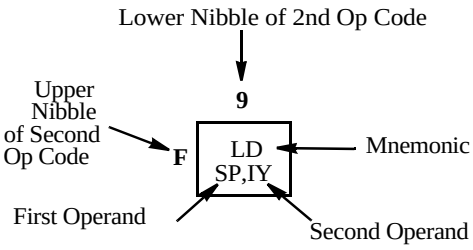


Table 12. Op Code Map (Fourth Byte After 0DDH, 0CBH, and d)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0							RLC (IX±d)								RRC (IX±d)	
1							RL (IX±d)								RR (IX±d)	
2							SLA (IX±d)								SRA (IX±d)	
3															SRL (IX±d)	
4							BIT 0, (IX±d)								BIT 1, (IX±d)	
5							BIT 2, (IX±d)								BIT 3, (IX±d)	
6							BIT 4, (IX±d)								BIT 5, (IX±d)	
7							BIT 6, (IX±d)								BIT 7, (IX±d)	
8							RES 0, (IX±d)								RES 1, (IX±d)	
9							RES 2, (IX±d)								RES 3, (IX±d)	
A							RES 4, (IX±d)								RES 5, (IX±d)	
B							RES 6, (IX±d)								RES 7, (IX±d)	
C							SET 0, (IX±d)								SET 1, (IX±d)	
D							SET 2, (IX±d)								SET 3, (IX±d)	
E							SET 4, (IX±d)								SET 5, (IX±d)	
F							SET 6, (IX±d)								SET 7, (IX±d)	

Note: d = signed 8-bit displacement

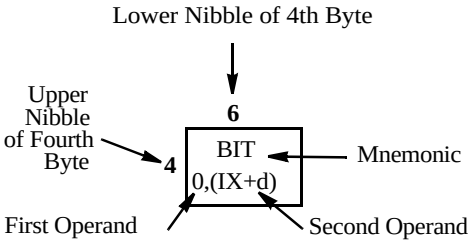
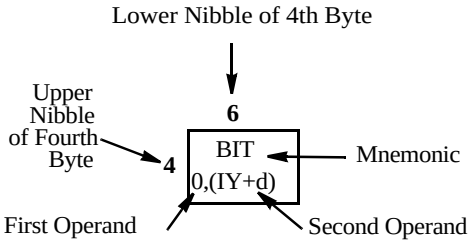


Table 13. Op Code Map (Fourth Byte After 0FDH, 0CBH, and d)

UPPER NIBBLE (HEX) \ LOWER NIBBLE (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0							RLC (IY±d)								RRC (IY±d)	
1							RL (IY±d)								RR (IY±d)	
2							SLA (IY±d)								SRA (IY±d)	
3															SRL (IY±d)	
4							BIT 0, (IY±d)								BIT 1, (IY±d)	
5							BIT 2, (IY±d)								BIT 3, (IY±d)	
6							BIT 4, (IY±d)								BIT 5, (IY±d)	
7							BIT 6, (IY±d)								BIT 7, (IY±d)	
8							RES 0, (IY±d)								RES 1, (IY±d)	
9							RES 2, (IY±d)								RES 3, (IY±d)	
A							RES 4, (IY±d)								RES 5, (IY±d)	
B							RES 6, (IY±d)								RES 7, (IY±d)	
C							SET 0, (IY±d)								SET 1, (IY±d)	
D							SET 2, (IY±d)								SET 3, (IY±d)	
E							SET 4, (IY±d)								SET 5, (IY±d)	
F							SET 6, (IY±d)								SET 7, (IY±d)	

Note: d = signed 8-bit displacement



Index

J

L

M

N

O

P

R

S

T

V

X

Z