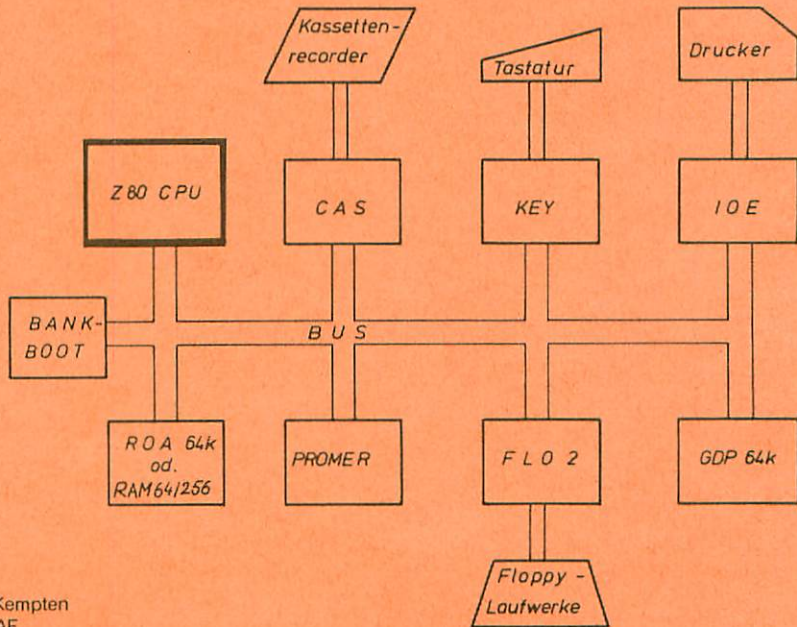


Der NDR-
Klein-Computer

CPU Z80



ger

Graf Elektronik Systeme GmbH

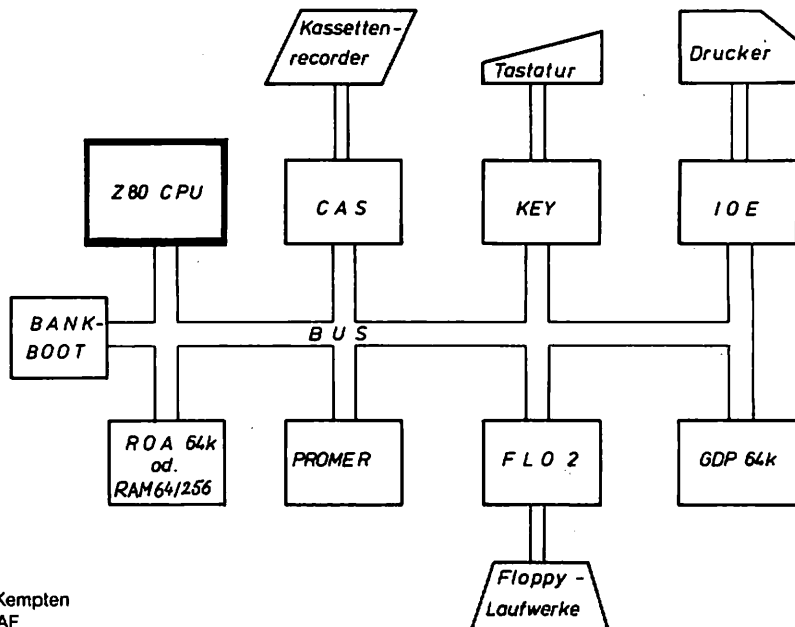
Magnusstraße 13 · Postfach 1610 · 8960 Kempten

Telefon (0831) 6211 · Teletex 831804 = GRAF

Der NDR-
Klein-Computer

CPU Z80

ger



Graf Elektronik Systeme GmbH
 Magnusstraße 13 · Postfach 1610 · 8960 Kempten
 Telefon (0831) 6211 · Teletex 831804 = GRAF

INHALTSVERZEICHNIS

Einführung	1
Stückliste	2
Bestückungsplan	4
Bestückungsanleitung	5
Test	6
Schaltungsbeschreibung	8
Schaltbild	9
Bauelemente	11
74 LS 245	11
74 LS 74	11
74 LS 121	12
CPU Z80A	13
Widerstandsfarbcodes	14

EINFÜHRUNG

Der NDR-Klein-Computer wird in der Fernsehserie "Mikroelektronik - Mikrocomputer selbstgebaut und programmiert" aufgebaut, erklärt und in Betrieb genommen. Diese Serie wird zunächst nur vom Norddeutschen Rundfunk, vom Sender Freies Berlin, von Radio Bremen und ab September 1984 vom Bayerischen Rundfunk ausgestrahlt. Es werden bald auch die Regionalsender anderer Bundesländer die Sendung in ihr Programm aufnehmen. Zur Serie gibt es einige Begleitmaterialien, sodaß es nicht unbedingt notwendig ist, die Fernsehserie gesehen zu haben, um den NDR-Klein-Computer zu bauen und zu begreifen:

- Buch: Rolf-Dieter Klein, "Mikrocomputer selbstgebaut und programmiert"; 2., neu bearbeitete und erweiterte Auflage, ISBN 3-7723-7162-0, DM 38,- ; erschienen im Franzis-Verlag, München.
Auf dieses Buch baut die NDR-Serie auf.

- mc-Hefte Nr.1 (Januar) bis Nr.6 (Juni) 1984: Erscheinen Anfang jedes Monats im Franzis-Verlag, München
Heftpreis: DM 6,- ; erhältlich im Buch- und Zeitschriftenhandel, sowie im Elektronikfachhandel.
Diese Hefte ergänzen -terminlich abgestimmt - die einzelnen Sendungen.

- mc-Sonderhefte: Diese Hefte, die vom Francis-Software-Service herausgegeben werden, behandeln jeweils ein spezielles Thema. Zur Zeit gibt es 6 Sonderhefte:

- | | |
|----------------------------|-------------------------|
| - Schaltpläne & Unterlagen | - 68008 Grundprogramme |
| - Z-80 Grundprogramme | - 68008 Aufbauprogramme |
| - Z-80 Aufbauprogramme | - 68008 PASCAL - S |

Außerdem gibt es noch kleine Heftchen zu den Programmen BASIC und GOSI

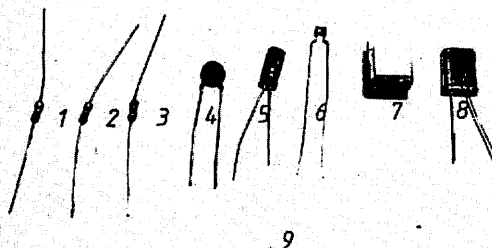
- BASIC für Z-80
- GOSI Einführung
- GOSI für Z-80

Diese Sonderhefte sind bei uns oder auch direkt beim Francis Verlag erhältlich.

- Videocassette 1/2: Lizenzierte Originalkassette für den privaten Gebrauch; Systeme: VHS, Beta, Video 2000
Auf diesen zwei Cassetten sind die 26 Folgen der Fernsehserie enthalten.

STÜCKLISTE

Stück	Bezeichnung	Nr. im Bild	
1	JC1	74 LS 04	
1	JC2	74 LS 00	
1	JC3	74 LS 74	
4	JC4, JC6, JC7, JC8	74 LS 245	
1	JC5	Z 80 A CPU	
1	JC9	74 LS 121	
4	S014	14-polige IC-Fassung	
4	S020	20-polige IC-Fassung	
1	S040	40-polige IC-Fassung	
8	R1,2,3,4,7,8,9,10	1 kOhm	1
1	R5	330 Ohm	2
1	R6	100 kOhm	3
1	C1	10 nF	4
2	C2, C4	10 uF	5
1	C3	100 nF	6
1	T1	Transistor	7
1	Q1	Quarz 4 MHz	8
1	St1	18- und 36-polige Steckerleiste	9
1	GES-Platine mit Lötstoplack		



4 SOCKEL 16POLIG



4 SOCKEL 20POLIG

2 SOCKEL
46POLIG

2 SOCKEL 80POLIG

CPU Z 80

Graf Elektronik Systeme GmbH



LS 04



74 121



LS 00



LS 74

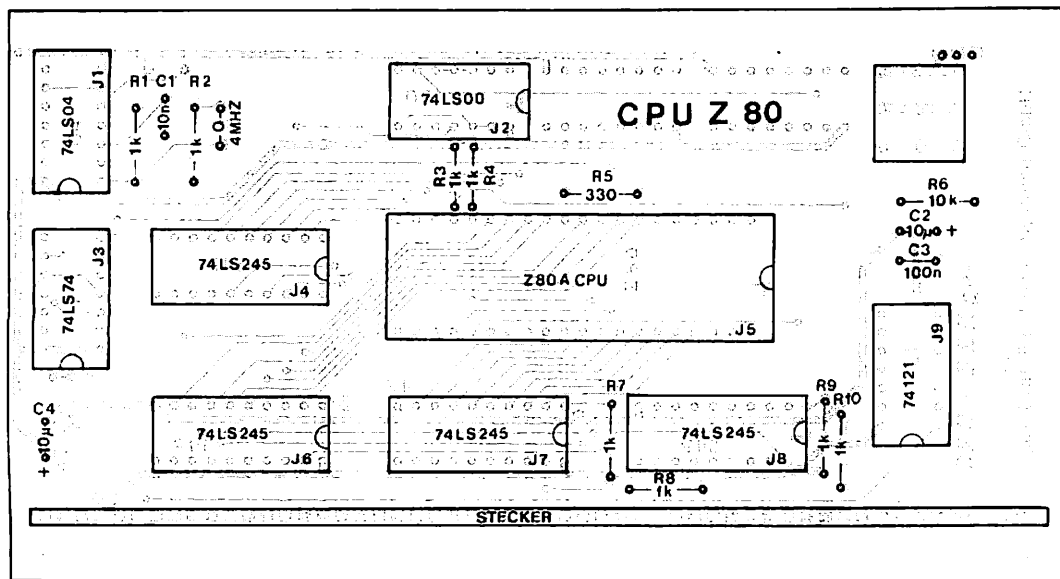


4 x 74LS 245



Graf Elektronik Systeme GmbH
 Magistrate 13 Postfach 1010
 D-960 Kempten
 Telefon 0931 8211
 Telex R31804 - GRAF
 von Telex 17 vorwählen

BESTÜCKUNGSPLAN



BESTÜCKUNGSANLEITUNG

Auf einer Seite der Platine steht der Hinweis "löts" (Lötseite); auf dieser Seite wird ausschließlich gelötet. Die Bauteile sind nur auf der anderen Seite aufzustecken.

Beim Einlöten der Bauelemente beginnt man am Besten mit der Steckerleiste. Es sollte darauf geachtet werden, daß die Stecker parallel zur Platine liegen, um gut auf die Busplatine gesteckt werden zu können. Dabei sollten zuerst die beiden äußeren Stifte und einer in der Mitte verlötet werden. Dann empfiehlt es sich nachzuschauen, ob die Stecker parallel zur Platine liegen und ob keine "Büchel" zwischen den verlöteten Stiften liegen.

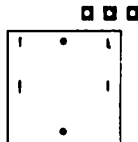
Als nächstes werden die IC-Sockel bestückt. Dabei muß darauf geachtet werden, daß die Fassungen richtig aufgesteckt werden. Im Bestückungsplan sind die Richtungen der Fassungen mit einer Kerbe gekennzeichnet. Sie muß mit der Richtung der Kerbe in der Fassung übereinstimmen. Außerdem ist die Lage der Fassungen auch auf der Bestückungsseite der Platine mit einem "I" gekennzeichnet. Die Kerbe muß hier in Richtung des Querbalkens des "I" liegen. Auf der CPU Z80-Platine ist nicht bei jedem IC dieses "I" aufgedruckt, sondern nur immer für eine IC-Reihe, denn alle IC's schauen in eine Richtung. Wo welche IC-Fassung hingehört, ist dem Bestückungsplan zu entnehmen.

Es sollten alle Fassungen auf einmal aufgesteckt werden und zum Verlöten umgedreht werden; dabei ist es hilfreich, wenn man beim Umdrehen die Fassungen mit einem Stück Karton auf die Platine andrückt. So wird erreicht, daß die Fassungen alle eben und gerade liegen. Beim Lüten sollten wiederum nur zwei Pins jeder Fassung (möglichst diagonal) verlötet werden. So können anschließend schräg liegende Fassungen noch problemlos korrigiert werden. Bevor die restlichen Pins verlötet werden, sollte noch ein letzter Kontrollblick auf die Bestückungsseite geworfen werden, ob die Fassungen richtig liegen und die Richtungen der Fassungen stimmen!

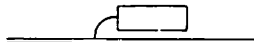
Die Keramikkondensatoren C1 und C3 sind ungepolt und können ohne Rücksicht auf bestimmte Anschlußrichtung aufgesteckt werden. Die auf der Lötseite herausstehenden Drähte sollten abgeknickt werden und kurz abgeschnitten, dadurch fällt der Kondensator beim Verlöten nicht heraus.

Die Elko's (Elektrolyt-Kondensator) C2 und C4 sind gepolt und dürfen auf keinen Fall falsch herum eingelötet werden. Der Minuspol ist auf dem Kondensator mit einem schwarzen Streifen gekennzeichnet, und auf der Platine sowie auf dem Bestückungsplan ist der Pluspol mit "+" gekennzeichnet.

Beim Einlöten des Tasters sollte nach untenstehender Skizze vorgegangen werden (von der Bestückungsseite aus gesehen). Die beiden schwarzen Stifte am Taster können beim Einstecken etwas klemmen. Läßt sich der Taster nicht stecken, können die beiden Kunststoffstifte abgezwickelt werden.



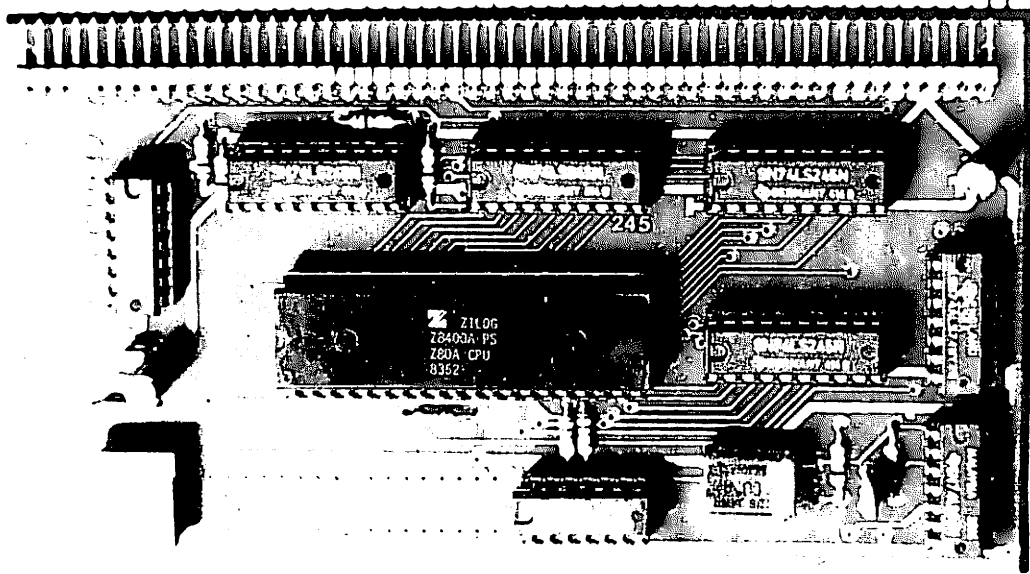
Der Quarz ist ungepolt, d.h. es muß beim Einlöten nicht auf die Polung geachtet werden. Er sollte liegend eingelötet werden (siehe Abbildung).



Die Widerstände sind ebenfalls ungepolt und können damit ohne auf die Polung zu achten eingelötet werden. Die Farbcodetabelle für die Widerstände finden sie hinten unter der Rubrik "Bauelemente".

TEST

1. Zuerst sollten alle Versorgungsspannungen aller IC's gemessen werden. Wo +5V und 0V liegen ist dem Pinning zu entnehmen (siehe Bauelemente)
2. Einstecken der IC's JC1 (74 04) und JC3 (74 LS 74). Vorsicht: nicht bei angelegter Spannung einstecken! Auf Richtung der IC's achten! An JC3/9 müssen nun 4MHz und an JC3/5 2MHz anliegen.
3. Einstecken der IC's JC9 (74 LS 121). Vorsicht wieder beim Einstecken des IC's. Wird nun der RESET-Testator gedrückt muß ein kurzer Impuls auf dem Oszilloskop sichtbar werden. Es ist ein einmaliger Impuls der nur kurz aufblinkt und daher nicht einfach zu erkennen. Der Impuls tritt in dem Moment auf in dem der Testator losgelassen wird.
4. Einstecken der restlichen IC's. Nicht bei angelegter Spannung einstecken und mehrmals kontrollieren, ob die IC's richtig herum eingesteckt sind. Damit wäre die Schaltung betriebsbereit.

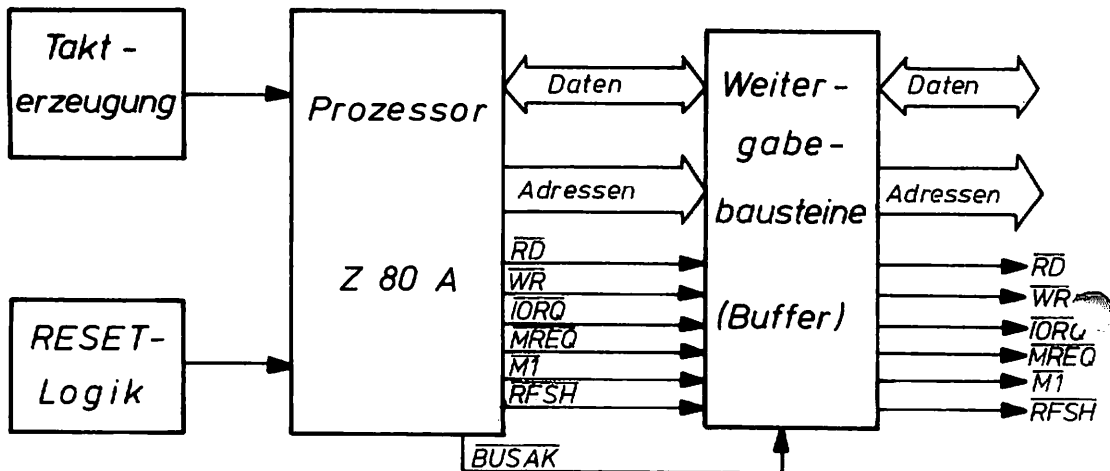


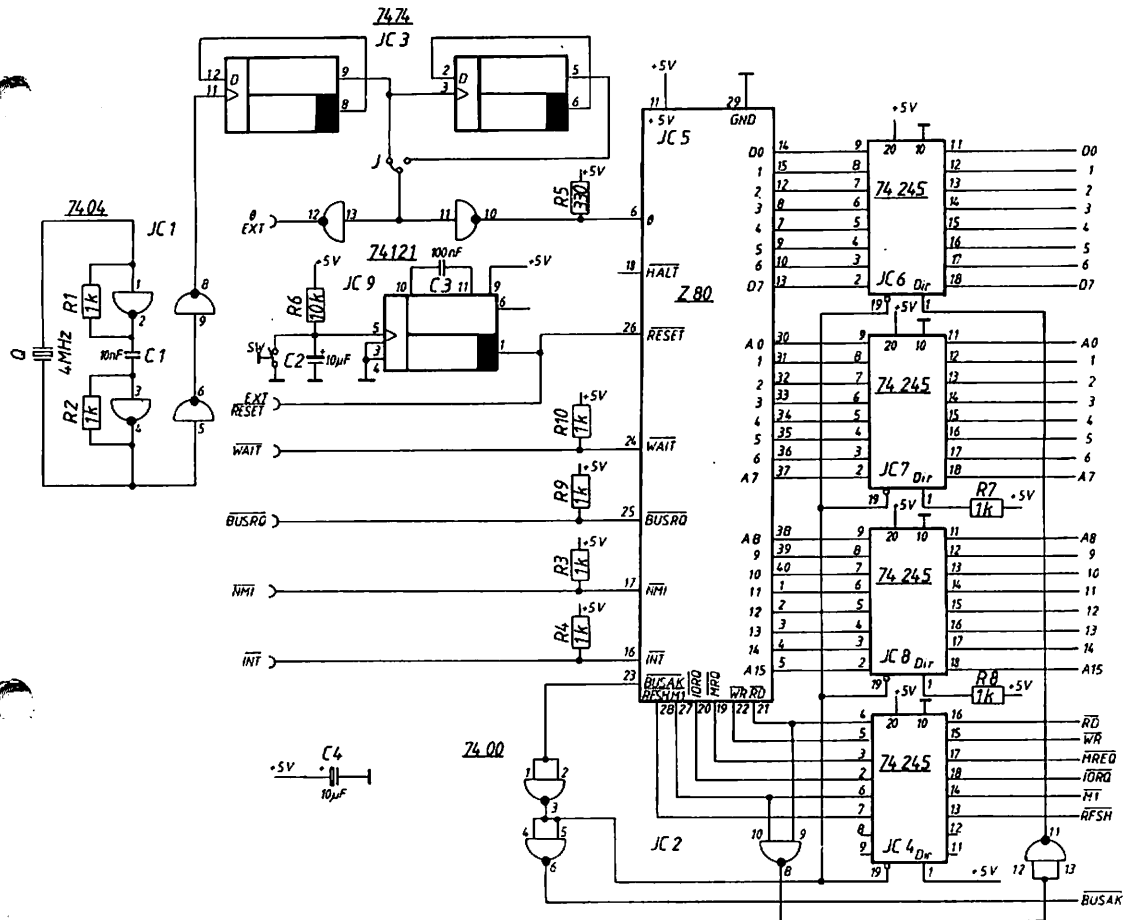
SCHALTUNGSBESCHREIBUNG

Die Schaltung lässt sich im wesentlichen in 4 Blöcke aufspalten.

1. Takterzeugung
2. Resetlogik
3. Weitergabebausteine
4. CPU Z80A

Blockschaltbild CPU Z 80



SCHALTBILD CPU Z 80

Takterzeugung

Der Takt wird, wie schon bei der SBC2 und der GDP64k mit 2 Invertern (JC1) erzeugt, deren Ausgänge mit einem 1 kOhm Widerstand rückgeführt werden auf den Eingang. Die beiden Schwingungserzeuger werden durch den 10 nF-Kondensator C1 miteinander verkoppelt und durch den Quarz Q1 auf 8 MHz stabilisiert. Die beiden D-Flip-Flops sind jeweils als Frequenzteiler 1:2 geschaltet. Die 8 MHz vom Frequenzgenerator werden beim 1. Flip-Flop auf 4 MHz und beim 2. Flip-Flop auf 2 MHz heruntergeteilt. Am JMP1 ist die Frequenz 4 MHz eingestellt.

RESET-Logik

Die RESET-Logik wird durch den Chip JC9 und den Taster T1 dargestellt. Der Baustein JC9 (74 LS 121) ist ein Monoflop. Wird der Taster T1 gedrückt entsteht am Ausgang ein einmaliger Impuls mit dem der "Program-Counter" (Programm-Zähler) der CPU rückgesetzt wird (RESET-Eingang JC5/26).

Weitergabebebausteine

Die Weitergabebebausteine sind das Bindeglied zwischen Mikroprozessor und dem Daten- bzw. Adressbus. Die Daten bzw. die Adressen werden an diesen Bausteinen abgeblockt (Buffer) und nur weitergegeben, wenn der Prozessor die erforderlichen Signale aussendet. Die Daten können hierbei in beiden Richtungen weitergeleitet werden, also vom Prozessor auf den Bus und umgekehrt, die Adressen allerdings nur vom Prozessor auf den Bus. Wann und in welche Richtung Daten weitergegeben werden sollen wird durch die HND-Verknüpfung der Signale RD und M1 (bestimmt die Richtung)(JC6/1), sowie dem Signal BUSAK* (JC6/19) gesteuert. Wann eine Adresse weitergegeben werden soll wird von dem Signal BUSAK* (Bus aktivieren) gesteuert (JC7/19 und JC8/19). Außerdem sind noch einige Steuerleitungen "gebuffert", die ebenfalls mit dem Signal BUSAK* weitergegeben werden (JC4/19). Da die Adressen und die Steuersignale nur in einer Richtung weitergegeben werden ist der Eingang DIR (Direction)(JC7/1, JC8/1 und JC4/1) fest an +5V gelegt.

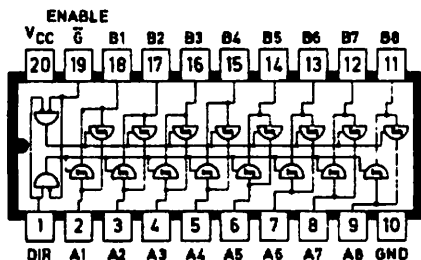
BAUELEMENTE

74 LS 245

8 Bus-Transceiver, Tri-State

- 20-poliger Chip
- 16 Ein- bzw. Ausgänge (A,B)
- 1 Eingang: G
- 1 Eingang: DIR

Es handelt sich hier um ein reines Tri-State-Element, d.h. er hat drei Funktionsmöglichkeiten. Er kann Daten von der A-Seite zur B-Seite oder umgekehrt durchschalten, oder aber er sperrt in beide Richtungen. Den sperrenden Zustand nennt man auch "hochohmig". Was das Chip machen soll, bestimmen die Signale DIR (Pin 1) und G (Pin 19).



Wahrheitstabelle

Enable G	Direction Control Dir.	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

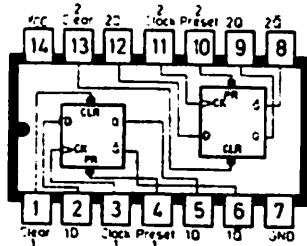
Positive Logik

74 LS 74

2 D-Flip-Flop mit Clear und Preset

Ist an D-Flip-Flop an "Preset" und "Clear" ein H-Signal, so wird das Flip-Flop bei Anlegen eines Taktsignals am "Clock"-Eingang, jeweils bei der positiven Flanke jedes Taktpulses abwechselnd gesetzt und rückgesetzt, d.h. die Ausgänge Q und Q* wechseln bei jeder positiven Taktfanke ihren Zustand. Liegt kein Takt an (Low-Signal) bleibt dieser Zustand erhalten (Speicherzustand).

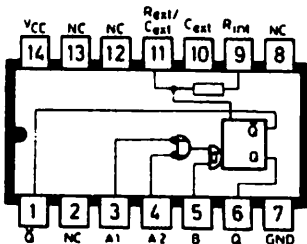
Will man diesen Zustand ändern, ohne einen Takt anzulegen, können die Signale an "Preset" und "Clear" verändert werden. Ist das Flip-Flop gesetzt (Q=H und Q*=L), kann es mit "Preset"=L und "Clear"=L zurückgesetzt werden (Q=L und Q*=H).



74 LS 121

Monoflop

Bei diesem Bauteil handelt es sich um ein Monoflop. Die Eingänge A1, A2 und B sind dazu da, das Monoflop zu triggern. Wie lange der Impuls am Ausgang des Monoflops (Pin 6 und 1) anliegen soll, wird durch ein RC-Glied festgelegt. Der Widerstand ist bereits im IC, während der Kondensator an die Pins 10 und 11 angeschlossen werden muß. Die Größe des Kondensators bestimmt dann die Länge des Ausgangsimpulses des Monoflops.



Wahrheitstabelle

Inputs				Outputs	
Preset	Clear	Clock	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q ₀	$\bar{Q}_0$

Positive Logik

* Dieser Zustand ist nicht stabil; d.h. er bleibt nicht erhalten, wenn Preset und/oder Clear inaktiv (High) werden.

Wahrheitstabelle

Inputs			Outputs	
A1	A2	B	Q	$\bar{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H	↓	H	⌊	⌋
↓	H	H	⌊	⌋
↓	↓	H	⌊	⌋
L	X	↑	⌊	⌋
X	L	↑	⌊	⌋

Positive Logik

Z80 CPU

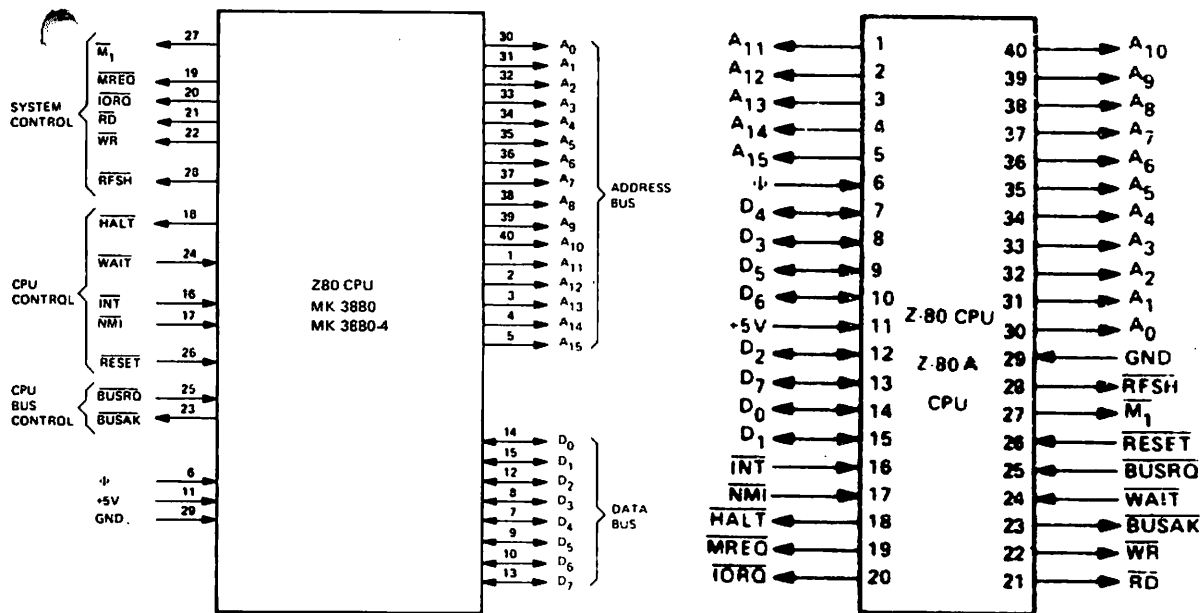


Abb. 9.1 Verschiedene Pin-Belegungen

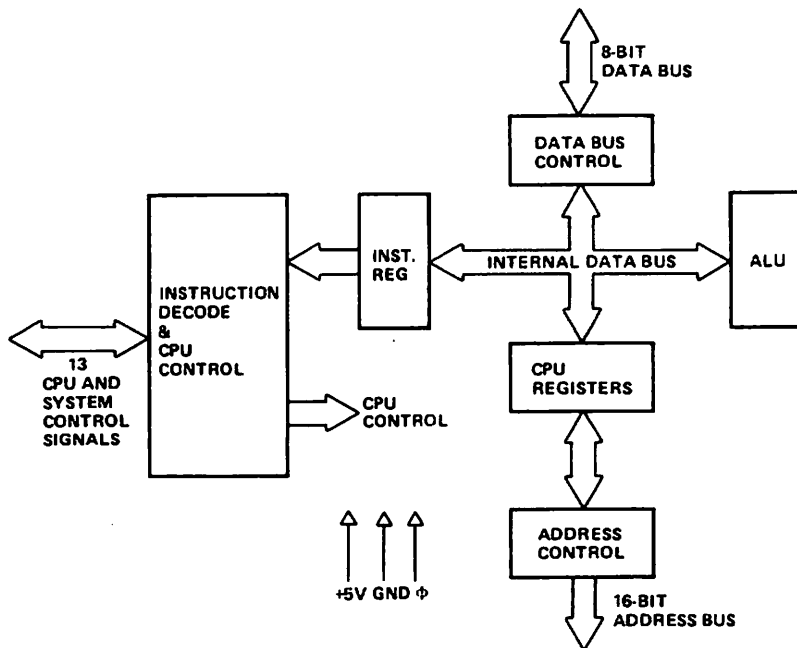
1.4 Praktischer Aufbau

Farbe	1. Ring = 1. Ziffer	2. Ring = 2. Ziffer	3. Ring = Anzahl der Nullen
schwarz	0	0	0
braun	1	1	1
rot	2	2	2
orange	3	3	3
gelb	4	4	4
grün	5	5	5
blau	6	6	6
violett	7	7	7
grau	8	8	8
weiß	9	9	9
gold	—	—	Wert · 0,1
silber	—	—	Wert · 0,01
4. Ring = Toleranz	gold ± 5 %	silber ± 10 %	kein Ring (fehlt) ± 20 %
Beispiel 1:	gelb 4	violett 7	rot 00 = 4,7 kΩ gold ± 5 %
Beispiel 2:	braun 1	schwarz 0	gold 0,1 = 1Ω silber ± 10 %
Beispiel 3:	braun 1	grün 5	silber 0,01 = 0,15Ω silber ± 10 %

Abb. Farbcode von Widerständen

2.0 Z-80 CPU ARCHITECTURE

A block diagram of the internal architecture of the Z-80 CPU is shown in figure 2.0-1. The diagram shows all of the major elements in the CPU and it should be referred to throughout the following description.



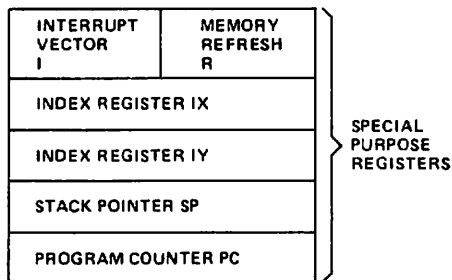
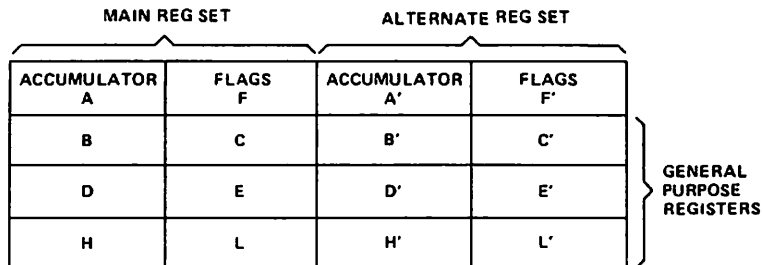
**Z-80 CPU BLOCK DIAGRAM
FIGURE 2.0-1**

2.1 CPU REGISTERS

The Z-80 CPU contains 208 bits of R/W memory that are accessible to the programmer. Figure 2.0-2 illustrates how this memory is configured into eighteen 8-bit registers and four 16-bit registers. All Z-80 registers are implemented using static RAM. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or in pairs as 16-bit registers. There are also two sets of accumulator and flag registers.

Special Purpose Registers

1. **Program Counter (PC).** The program counter holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.
2. **Stack Pointer (SP).** The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack from specific CPU registers or popped off of the stack into specific CPU registers through the execution of PUSH and POP instructions. The data popped from the stack is always the last data pushed onto it. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.



Z-80 CPU REGISTER CONFIGURATION
FIGURE 2.0-2

3. **Two Index Registers (IX & IY).** The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer. This mode of addressing greatly simplifies many types of programs, especially where tables of data are used.
4. **Interrupt Page Address Register (I).** The Z-80 CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I Register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrupt routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.
5. **Memory Refresh Register (R).** The Z-80 CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. Seven bits of this 8 bit register are automatically incremented after each instruction fetch. The eighth bit will remain as programmed as the result of an LD R, A instruction. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer. During refresh, the contents of the I register are placed on the upper 8 bits of the address bus.

Accumulator and Flag Registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8 or 16-bit operations, such as indicating whether or not the result of an operation is equal to zero. The programmer selects the accumulator and flag pair that he wishes to work with with a single exchange instruction so that he may easily work with either pair.

General Purpose Registers

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or as 16-bit register pairs by the programmer. One set is called BC, DE and HL while the complementary set is called BC', DE' and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator/flag register may be reserved for handling this very fast routine. Only a simple exchange commands need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

2.2 ARITHMETIC & LOGIC UNIT (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus. The type of functions performed by the ALU include:

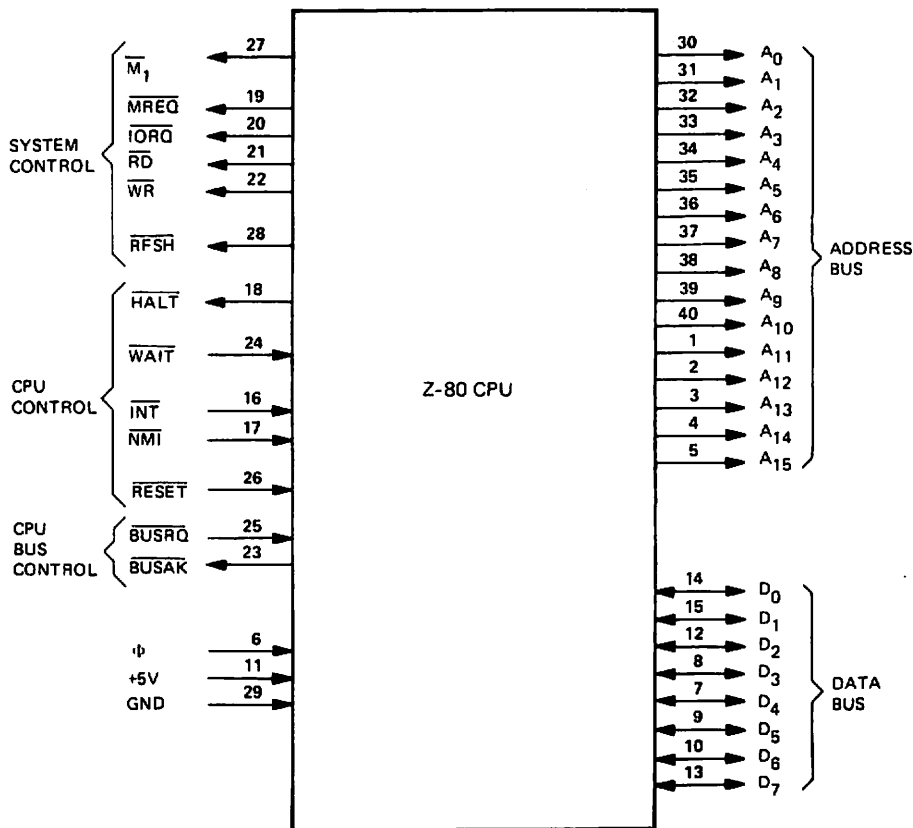
Add	Left or right shifts or rotates (arithmetic and logical)
Subtract	Increment
Logical AND	Decrement
Logical OR	Set bit
Logical Exclusive OR	Reset bit
Compare	Test bit

2.3 INSTRUCTION REGISTER AND CPU CONTROL

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control sections performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, control the ALU and provide all required external control signals.

3.0 Z-80 CPU PIN DESCRIPTION

The Z-80 CPU is packaged in an industry standard 40 pin Dual In-Line Package. The I/O pins are shown in figure 3.0-1 and the function of each is described below.



Z-80 PIN CONFIGURATION
FIGURE 3.0-1

A_0-A_{15}
(Address Bus)

Tri-state output, active high. A_0-A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges. I/O addressing uses the 8 lower address bits to allow the user to directly select up to 256 input or 256 output ports. A_0 is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address.

D_0-D_7
(Data Bus)

Tri-state input/output, active high. D_0-D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

$\overline{M}_1$
(Machine Cycle one)

Output, active low. $\overline{M}_1$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes, $\overline{M}_1$ is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH or FDH. $\overline{M}_1$ also occurs with $\overline{IORQ}$ to indicate an interrupt acknowledge cycle.

$\overline{MREQ}$
(Memory Request)

Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{\text{IORQ}}$
(Input/Output Request)

Tri-state output, active low. The $\overline{\text{IORQ}}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{\text{IORQ}}$ signal is also generated with an $\overline{\text{M1}}$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during M_1 time while I/O operations never occur during M_1 time.

$\overline{\text{RD}}$
(Memory Read)

Tri-state output, active low. $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{\text{WR}}$
(Memory Write)

Tri-state output, active low. $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

$\overline{\text{RFSH}}$
(Refresh)

Output, active low. $\overline{\text{RFSH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{\text{MREQ}}$ signal should be used to do a refresh read to all dynamic memories.

$\overline{\text{HALT}}$
(Halt state)

Output, active low. $\overline{\text{HALT}}$ indicates that the CPU has executed a HALT software instruction and is awaiting either a non maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

WAIT
(Wait)

Input, active low. WAIT indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.

INT
(Interrupt Request)

Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled and if the BUSRQ signal is not active. When the CPU accepts the interrupt, an acknowledge signal (IORQ during M_1 time) is sent out at the beginning of the next instruction cycle. The CPU can respond to an interrupt in three different modes that are described in detail in section 5.4 (CPU Control Instructions).

NMI
(Non Maskable
Interrupt)

Input, negative edge triggered. The non maskable interrupt request line has a higher priority than INT and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. NMI automatically forces the Z-80 CPU to restart to location 0066H. The program counter is automatically saved in the external stack so that the user can return to the program that was interrupted. Note that continuous WAIT cycles can prevent the current instruction from ending, and that a BUSRQ will override a NMI.

RESET

Input, active low. RESET forces the program counter to zero and initializes the CPU. The CPU initialization includes:

- 1) Disable the interrupt enable flip-flop
- 2) Set Register I = 00_H
- 3) Set Register R = 00_H
- 4) Set Interrupt Mode 0

During reset time, the address bus and data bus go to a high impedance state and all control output signals go to the inactive state.

BUSRQ

(Bus Request)

Input, active low. The bus request signal is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When BUSRQ is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.

BUSAK

(Bus Acknowledge)

Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

Φ

Single phase TTL level clock which requires only a 330 ohm pull-up resistor to +5 volts to meet all clock requirements.

Mnemonic	Symbolic Operation	Flags						OP-Code	No. of Bytes	No. of M Cycles	No. of T Cycles	Comments
		C	Z	P/V	S	N	H					
LD r, r'	r ← r'	•	•	•	•	•	•	01 r r'	1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	•	•	•	•	00 r 110	2	2	7	000 B
								← n →				001 C
								← d →				101 L
LD r, (HL)	r ← (HL)	•	•	•	•	•	•	01 r 110	1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	•	•	•	•	11 011 101	3	5	19	011 E
								01 r 110				100 H
								← d →				101 L
LD r, (IY+d)	r ← (IY+d)	•	•	•	•	•	•	11 111 101	3	5	19	111 A
								01 r 110				
								← d →				
LD (HL), r	(HL) ← r	•	•	•	•	•	•	01 110 r	1	2	7	
LD (IX+d), r	(IX+d) ← r	•	•	•	•	•	•	11 011 101	3	5	19	
								01 110 r				
								← d →				
LD (IY+d), r	(IY+d) ← r	•	•	•	•	•	•	11 111 101	3	5	19	
								01 110 r				
								← d →				
LD (HL), n	(HL) ← n	•	•	•	•	•	•	00 110 110	2	3	10	
								← n →				
								← d →				
LD (IX+d), n	(IX+d) ← n	•	•	•	•	•	•	11 011 101	4	5	19	
								00 110 110				
								← d →				
LD (IY+d), n	(IY+d) ← n	•	•	•	•	•	•	11 111 101	4	5	19	
								00 110 110				
								← d →				
								← n →				

Mnemonic	Symbolic Operation	Flags						OP-Code			No. of Bytes	No. of M Cycles	No. of T Cycles	Comments
		C	Z	P/V	S	N	H	76	543	210				
LD A, (BC)	A ← (BC)	•	•	•	•	•	•	00	001	010	1	2	7	
LD A, (DE)	A ← (DE)	•	•	•	•	•	•	00	011	010	1	2	7	
LD A, (nn)	A ← (nn)	•	•	•	•	•	•	00	111	010	3	4	13	
								←	n	→				
								←	n	→				
LD (BC), A	(BC) ← A	•	•	•	•	•	•	00	000	010	1	2	7	
LD (DE), A	(DE) ← A	•	•	•	•	•	•	00	010	010	1	2	7	
LD (nn), A	(nn) ← A	•	•	•	•	•	•	00	110	010	3	4	13	
								←	n	→				
								←	n	→				
LD A, I	A ← I	•	‡	IFF	‡	0	0	11	101	101	2	2	9	
								01	010	111				
LD A, R	A ← R	•	‡	IFF	‡	0	0	11	101	101	2	2	9	
								01	011	111				
LD I, A	I ← A	•	•	•	•	•	•	11	101	101	2	2	9	
								01	000	111				
LD R, A	R ← A	•	•	•	•	•	•	11	101	101	2	2	9	
								01	001	111				

Notes: r, r' means any of the registers A, B, C, D, E, H, L

IFF: the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,

‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	V	S	N	H	76	543	210				
LD dd, nn	dd ← nn	•	•	•	•	•	•	00	dd0	001	3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	•	•	•	•	11	011	101	4	4	14	
LD IY, nn	IY ← nn	•	•	•	•	•	•	11	111	101	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	•	•	•	•	00	101	010	3	5	16	
LD dd, (nn)	dd _H ← (nn+1) dd _L ← (nn)	•	•	•	•	•	•	11	101	101	4	6	20	
LD IX, (nn)	IX _H ← (nn+1) IX _L ← (nn)	•	•	•	•	•	•	11	011	101	4	6	20	
LD IY, (nn)	IY _H ← (nn+1) IY _L ← (nn)	•	•	•	•	•	•	11	111	101	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	•	•	•	•	00	100	010	3	5	16	
LD (nn), dd	(nn+1) ← dd _H (nn) ← dd _L	•	•	•	•	•	•	11	101	101	4	6	20	
LD (nn), IX	(nn+1) ← IX _H (nn) ← IX _L	•	•	•	•	•	•	11	011	101	4	6	20	
LD (nn), IY	(nn+1) ← IY _H (nn) ← IY _L	•	•	•	•	•	•	11	111	101	4	6	20	

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	V	S	N	H	76 543 210				
LD SP, HL	SP ← HL	•	•	•	•	•	•	11 111 001	1	1	6	
LD SP, IX	SP ← IX	•	•	•	•	•	•	11 011 101	2	2	10	
								11 111 001				
LD SP, IY	SP ← IY	•	•	•	•	•	•	11 111 101	2	2	10	
								11 111 001				
PUSH qq	(SP-2) ← qq _L	•	•	•	•	•	•	11 qq0 101	1	3	11	qq Pair
	(SP-1) ← qq _H											00 BC
PUSH IX	(SP-2) ← IX _L	•	•	•	•	•	•	11 011 101	2	4	15	01 DE
	(SP-1) ← IX _H							11 100 101				10 HL
PUSH IY	(SP-2) ← IY _L	•	•	•	•	•	•	11 111 101	2	4	15	11 AF
	(SP-1) ← IY _H							11 100 101				
POP qq	qq _H ← (SP+1)	•	•	•	•	•	•	11 qq0 001	1	3	10	
	qq _L ← (SP)											
POP IX	IX _H ← (SP+1)	•	•	•	•	•	•	11 011 101	2	4	14	
	IX _L ← (SP)							11 100 001				
POP IY	IY _H ← (SP+1)	•	•	•	•	•	•	11 111 101	2	4	14	
	IY _L ← (SP)							11 100 001				

Notes: dd is any of the register pairs BC, DE, HL, SP

qq is any of the register pairs AF, BC, DE, HL

(PAIR)_H, (PAIR)_L refer to high order and low order eight bits of the register pair respectively.

E.g. BC_L = C, AF_H = A

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE, HL	DE ← HL	•	•	•	•	•	•	11	101	011	1	1	4	Register bank and auxiliary register bank exchange
EX AF, AF'	AF ← AF'	•	•	•	•	•	•	00	001	000	1	1	4	
EXX	BC DE HL ← BC' DE' HL'	•	•	•	•	•	•	11	011	001	1	1	4	
EX (SP), HL	H ← (SP+1) L ← (SP)	•	•	•	•	•	•	11	100	011	1	5	19	Register bank and auxiliary register bank exchange
EX (SP), IX	IX _H ← (SP+1) IX _L ← (SP)	•	•	•	•	•	•	11	011	101	2	6	23	
EX (SP), IY	IY _H ← (SP+1) IY _L ← (SP)	•	•	•	•	•	•	11	111	101	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	1	•	0	0	11	101	101	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
								10	100	000				
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	0	•	0	0	11	101	101	2	5	21	If BC ≠ 0 If BC = 0
								10	110	000	2	4	16	
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC+1	•	•	1	•	0	0	11	101	101	2	4	16	If BC ≠ 0 If BC = 0
								10	101	000				
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC+1 Repeat until BC = 0	•	•	0	•	0	0	11	101	101	2	5	21	
								10	111	000	2	4	16	
				0	1									

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
CPI	A ← (HL)	•	‡	‡	‡	‡	‡	11	101	101	2	4	16	
	HL ← HL+1							10	100	001				
	BC ← BC-1													
CPIR			②	①										
	A ← (HL)	•	‡	‡	‡	‡	‡	11	101	101	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL+1							10	110	001	2	4	16	If BC = 0 or A = (HL)
	BC ← BC-1													
	Repeat until A = (HL) or BC = 0													
CPD			②	①										
	A ← (HL)	•	‡	‡	‡	‡	‡	11	101	101	2	4	16	
	HL ← HL-1							10	101	001				
CPDR			②	①										
	A ← (HL)	•	‡	‡	‡	‡	‡	11	101	101	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL-1							10	111	001	2	4	16	If BC = 0 or A = (HL)
	BC ← BC-1													
	Repeat until A = (HL) or BC = 0													

Notes: ① P-V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1

② Z flag is 1 if A = (HL), otherwise Z = 0.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.

‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
ADD A, r	A ← A + r	†	†	V	†	0	†	10	000	r	1	1	4	r Reg.
ADD A, n	A ← A + n	†	†	V	†	0	†	11	000	110	2	2	7	000 B
										← n →				001 C
ADD A, (HL)	A ← A + (HL)	†	†	V	†	0	†	10	000	110	1	2	7	010 D
ADD A, (IX+d)	A ← A + (IX+d)	†	†	V	†	0	†	11	011	101	3	5	19	011 E
										10 000 110				100 H
										← d →				101 L
ADD A, (IY+d)	A ← A + (IY+d)	†	†	V	†	0	†	11	111	101	3	5	19	111 A
										10 000 110				
										← d →				
ADC A, s	A ← A + s + CY	†	†	V	†	0	†		001					s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction
SUB s	A ← A - s	†	†	V	†	1	†		010					
SBC A, s	A ← A - s - CY	†	†	V	†	1	†		011					
AND s	A ← A ∧ s	0	†	P	†	0	1		100					
OR s	A ← A ∨ s	0	†	P	†	0	0		110					The indicated bits replace the 000 in the ADD set above.
XOR s	A ← A ⊕ s	0	†	P	†	0	0		101					
CP s	A - s	†	†	V	†	1	†		111					
INC r	r ← r + 1	•	†	V	†	0	†	00	r	100	1	1	4	
INC (HL)	(HL) ← (HL) + 1	•	†	V	†	0	†	00	110	100	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d) + 1	•	†	V	†	0	†	11	011	101	3	6	23	
										00 110 100				
										← d →				
INC (IY+d)	(IY+d) ← (IY+d) + 1	•	†	V	†	0	†	11	111	101	3	6	23	
										00 110 100				
										← d →				
DEC m	m ← m - 1	•	†	V	†	1	†			101				m is any of r, (HL), (IX+d), (IY+d) as shown for INC. Same format and states as INC. Replace 100 with 101 in OP code.

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow. P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, † = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	‡	‡	P	‡	•	‡	00	100	111	1	1	4	Decimal adjust accumulator
CPL	$A \leftarrow \bar{A}$	•	•	•	•	1	1	00	101	111	1	1	4	Complement accumulator (one's complement)
NEG	$A \leftarrow 0 - A$	‡	‡	V	‡	1	‡	11	101	101	2	2	8	Negate acc. (two's complement)
CCF	$CY \leftarrow \bar{CY}$	‡	•	•	•	0	X	00	111	111	1	1	4	Complement carry flag
SCF	$CY \leftarrow 1$	1	•	•	•	0	0	00	110	111	1	1	4	Set carry flag
NOP	No operation	•	•	•	•	•	•	00	000	000	1	1	4	
HALT	CPU halted	•	•	•	•	•	•	01	110	110	1	1	4	
DI	$IFF \leftarrow 0$	•	•	•	•	•	•	11	110	011	1	1	4	
EI	$IFF \leftarrow 1$	•	•	•	•	•	•	11	111	011	1	1	4	
IM 0	Set interrupt mode 0	•	•	•	•	•	•	11	101	101	2	2	8	
IM 1	Set interrupt mode 1	•	•	•	•	•	•	11	101	101	2	2	8	
								01	010	110				
IM2	Set interrupt mode 2	•	•	•	•	•	•	11	101	101	2	2	8	
								01	011	110				

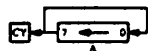
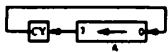
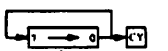
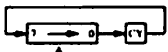
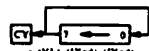
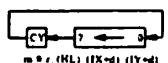
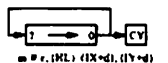
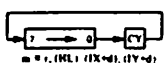
Notes: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.

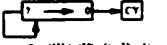
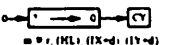
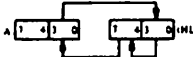
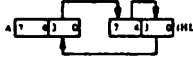
Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
ADD HL, ss	HL ← HL + ss	‡	•	•	•	0	X	00	ss1	001	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL + ss + CY	‡	‡	V	‡	0	X	11	101	101	2	4	15	
SBC HL, ss	HL ← HL - ss - CY	‡	‡	V	‡	1	X	11	101	101	2	4	15	
ADD IX, pp	IX ← IX + pp	‡	•	•	•	0	X	11	011	101	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	‡	•	•	•	0	X	11	111	101	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	•	•	•	•	•	•	00	ss0	011	1	1	6	
INC IX	IX ← IX + 1	•	•	•	•	•	•	11	011	101	2	2	10	
INC IY	IY ← IY + 1	•	•	•	•	•	•	11	111	101	2	2	10	
DEC ss	ss ← ss - 1	•	•	•	•	•	•	00	ss1	011	1	1	6	
DEC IX	IX ← IX - 1	•	•	•	•	•	•	11	011	101	2	2	10	
DEC IY	IY ← IY - 1	•	•	•	•	•	•	11	111	101	2	2	10	

Notes: ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code				No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210					
RLCA		†	•	•	•	0	0	00	000	111	1	1	4	Rotate left circular accumulator	
RLA		†	•	•	•	0	0	00	010	111	1	1	4	Rotate left accumulator	
RRCA		†	•	•	•	0	0	00	001	111	1	1	4	Rotate right circular accumulator	
RRA		†	•	•	•	0	0	00	011	111	1	1	4	Rotate right accumulator	
RLC r	 r, (HL), (IX+d), (IY+d)	†	†	P	†	0	0	11	001	011	2	2	8	Rotate left circular register r	
RLC (HL)		†	†	P	†	0	0	11	001	011	2	4	15	r Reg. 000 B	
RLC (IX+d)		†	†	P	†	0	0	11	000	110	4	6	23	001 C 010 D 011 E 100 H 101 L 111 A	
		†	†	P	†	0	0	11	011	101					
		†	†	P	†	0	0	11	001	011					
		†	†	P	†	0	0	11	001	011					
RLC (IY+d)	†	†	P	†	0	0	11	111	101	4	6	23			
RL m	 m = r, (HL), (IX+d), (IY+d)	†	†	P	†	0	0	00	000	110				Instruction format and states are as shown for RLC m. To form new OP-code replace 000 of RLC m with shown code	
RRC m	 m = r, (HL), (IX+d), (IY+d)	†	†	P	†	0	0	00	001						
RR m	 m = r, (HL), (IX+d), (IY+d)	†	†	P	†	0	0	00	011						

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H					
SLA m	 m ← (HL) << 1 (CY ← d)	?	?	P	?	0	0	100				
SRA m	 m ← (HL) >> 1 (CY ← d)	?	?	P	?	0	0	101				
SRL m	 m ← (HL) >> 1 (CY ← d)	?	?	P	?	0	0	111				
RLD	 A ← (A >> 4) (HL << 4), HL ← (HL >> 4) (A << 4)	•	?	P	?	0	0	11 101 101 01 101 111	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected
RRD	 A ← (A >> 4) (HL << 4), HL ← (HL >> 4) (A << 4)	•	?	P	?	0	0	11 101 101 01 100 111	2	5	18	

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
? = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments	
		C	Z	$\overline{V}$	S	N	H	76	543	210					
BIT b, r	$Z \sim \overline{r}_b$	•	†	X	X	0	1	11	001	011	2	2	8	r	Reg.
								01	b	r				000	B
BIT b, (HL)	$Z \sim \overline{(HL)}_b$	•	†	X	X	0	1	11	001	011	2	3	12	001	C
								01	b	110				010	D
BIT b, (IX+d)	$Z \sim \overline{(IX+d)}_b$	•	†	X	X	0	1	11	011	101	4	5	20	011	E
								11	001	011				100	H
								←	d	→				101	L
								01	b	110				111	A
BIT b, (IY+d)	$Z \sim \overline{(IY+d)}_b$	•	†	X	X	0	1	11	111	101	4	5	20	b	Bit Tested
								11	001	011				000	0
								←	d	→				001	1
								01	b	110				010	2
														011	3
														100	4
														101	5
														110	6
														111	7
SET b, r	$r_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	2	2	8		
								11	b	r					
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	2	4	15		
								11	b	110					
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	11	011	101	4	6	23		
								11	001	011					
								←	d	→					
								11	b	110					
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	11	111	101	4	6	23		
								11	001	011					
								←	d	→					
								11	b	110					

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
RES b, m	$s_b \leftarrow 0$ $m \equiv r, (HL),$ $(IX+d),$ $(IY+d)$									110				To form new OP-code replace 11 of SET b,m with 10. Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) or location a.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments	
		C	Z	P/V	S	N	H	76	543	210					
JP nn	PC ← nn	•	•	•	•	•	•	11	000	011	3	3	10		
								←	n	→					
								←	n	→					
JP cc, nn	If condition cc is true PC ←nn, otherwise continue	•	•	•	•	•	•	11	cc	010	3	3	10	cc	Condition
								←	n	→				000	NZ non zero
								←	n	→				001	Z zero
								←	n	→				010	NC non carry
								←	n	→				011	C carry
														100	PO parity odd
														101	PE parity even
														110	P sign positive
														111	M sign negative
JR e	PC ← PC + e	•	•	•	•	•	•	00	011	000	2	3	12		
								←	e-2	→					
JR C, e	If C = 0, continue	•	•	•	•	•	•	00	111	000	2	2	7	If condition not met	
	If C = 1, PC ← PC+e							←	e-2	→	2	3	12	If condition is met	
JR NC, e	If C = 1, continue	•	•	•	•	•	•	00	110	000	2	2	7	If condition not met	
	If C = 0, PC ← PC + e							←	e-2	→	2	3	12	If condition is met	
JR Z, e	If Z = 0 continue	•	•	•	•	•	•	00	101	000	2	2	7	If condition not met	
	If Z = 1, PC ← PC + e							←	e-2	→	2	3	12	If condition is met	
JR NZ, e	If Z = 1, continue	•	•	•	•	•	•	00	100	000	2	2	7	If condition not met	
	If Z = 0, PC ← PC + e							←	e-2	→	2	3	12	If condition met	

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
JP (HL)	PC ← HL	•	•	•	•	•	•	11	101	001	1	1	4	
JP (IX)	PC ← IX	•	•	•	•	•	•	11	011	101	2	2	8	
								11	101	001				
JP (IY)	PC ← IY	•	•	•	•	•	•	11	111	101	2	2	8	
								11	101	001				
DJNZ,e	B ← B-1 If B = 0, continue	•	•	•	•	•	•	00	010	000	2	2	8	If B = 0
								←	e-2	→				
	If B ≠ 0, PC ← PC + e										2	3	13	If B ≠ 0

Notes: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range <-126, 129>

e-2 in the op-code provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
CALL nn	(SP-1)→PC _H (SP-2)→PC _L PC←nn	•	•	•	•	•	•	11	001	101	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	•	•	•	•	11	cc	100	3	3	10	If cc is false
											← n →	5	17	If cc is true
RET	PC _L ←(SP) PC _H ←(SP+1)	•	•	•	•	•	•	11	001	001	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	•	•	•	•	11	cc	000	1	1	5	If cc is false
											1	3	11	If cc is true
RETI	Return from interrupt	•	•	•	•	•	•	11	101	101	2	4	14	
RETN	Return from non maskable interrupt	•	•	•	•	•	•	11	101	101	2	4	14	
RST p	(SP-1)→PC _H (SP-2)→PC _L PC _H ←0 PC _L ←P	•	•	•	•	•	•	11	t	111	1	3	11	

cc	Condition	
000	NZ	non zero
001	Z	zero
010	NC	non carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	P
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown
 ‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
IN A, (n)	A ← (n)	•	•	•	•	•	•	11	011	011	2	3	11	n to A ₀ ~ A ₇ Acc to A ₈ ~ A ₁₅
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	•	†	P	†	0	†	11	101	101	2	3	12	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	①	†	X	X	1	X	11	101	101	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	•	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	110	010	2	4 (If B = 0)	16	
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	①	†	X	X	1	X	11	101	101	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 Repeat until B = 0	•	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	111	010	2	4 (If B = 0)	16	
OUT (n), A	(n) ← A	•	•	•	•	•	•	11	010	011	2	3	11	n to A ₀ ~ A ₇ Acc to A ₈ ~ A ₁₅
OUT (C), r	(C) ← r	•	•	•	•	•	•	11	101	101	2	3	12	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
		①	†	X	X	1	X	01	r	001				
OUTI	(C) ← (HL) B ← B - 1 HL ← HL + 1	•	†	X	X	1	X	11	101	101	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	100	011				

Mnemonic	Symbolic Operation	Flags						(Op-Code)			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P	S	N	H	76	543	210				
OTIR	(C) ← (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	•	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	110	011	2	4 (If B = 0)	16	
OUTD	(C) ← (HL) B ← B - 1 HL ← HL - 1	•	①	X	X	1	X	11	101	101	2	4	16	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	101	011				
OTDR	(C) ← (HL) B ← B - 1 HL ← HL - 1 Repeat until B = 0	•	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅
								10	111	011	2	4 (If B = 0)	16	

Notes: ① If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

