

00240	.....		
00250	.....		
00260	.....		
401E	00270	VIDOR!	EQU
FF00	2A1E40	00280	ANF
FF03	220EFF	00290	LD
		00300	LD
FF06	2100FF	00310	LD
FF09			LD
FF0C			ET

**BLOOMP**

401EH  
HL, (VIDORI)  
(PROTON+1),  
HL, PROTON  
(VIDORI), HL



**Z80**

FF00			ALL	0000H
FF10			CP	95
FF12			TR	Z, ENDE
FF14			PUSH	BC
FF18			LD	C, A
		00420		
FF16	2A5837	00430	PRINT	LD
FF19	0B7F	00440		BIT
FF1B	C216FF	00450		JP

# Handbuch

		00510		
FF23	C9	00520	ENDE	RET
FF00		00530		END
				ANF
00000	TOTAL	ERRORS		
ANF	FF00	00280	00530	
ENDE	FF23	00520	00390	
PRINT	FF16	00430	00450	
PROTON	FF00	00370	00290	00310



00240 ; \*\*\*\*\*  
 00250 ; \*\*\*\*\*  
 00260 ; \*\*\*\*\*

B **ELOOMP**

1E 00270 VIDDRI  
 00 2A1E40 00280 ANF  
 03 220EFF 00290  
 00300  
 06 2100FF 00310

EQU 401EH  
 LD HL, (VIDDRI)  
 LD (PROTON+1),  
 LD HL, PROTON  
 LD (VIDDRI), HL  
 RET



DRUCKERAUSGABE  
**Z80**

00420  
 16 3AE237 00430 PRINT  
 19 0B15 00440  
 1B C216FF 00450

CALL 0000H  
 CP 95  
 JR Z, ENDE  
 PUSH BC  
 LD C, A  
 LD A, (37E8H)  
 BIT 7, A  
 JP NZ, PRINT

**ASSEMBLER**

# Handbuch

00510  
 23 C9 00520 ENDE  
 00 00530

RET  
 END ANF

1000 TOTAL ERRORS  
 IF FF00 00280 00530  
 DE FF23 00520 00390  
 INT FF16 00430 00450  
 OTON FF00 00370 00290 00310



8 Ac 73

ISBN 3-921682-74-6

Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

**COPYRIGHT BY ING. W. HOFACKER** © 1980, Postfach 75 437, 8000 München 75

1. Auflage 1980

Gedruckt in der Bundesrepublik Deutschland — Printed in West-Germany — Imprime'en RFA.



# Inhaltsverzeichnis

40		
FF		
FF	8 bit-Ladebefehle	1
	LD r, r'	3
	LD r, n	5
FF	LD r, (HL)	7
FF	LD r, (IX+d)	9
FF	LD r, (IY+d)	11
	LD (HL), r	13
	LD (IX+d), r	15
	LD (IY+d), r	17
	LD (HL), n	19
	LD (IX+d), n	21
	LD (IY+d), n	23
FF	LD A, (BC)	25
FF	LD A, (DE)	27
FF	LD A, (nn)	29
FF	LD (BC), A	31
FF	LD (DE), A	33
	LD (nn), A	35
FF	LD A, I	37
FF	LD A, R	39
FF	LD I, A	41
FF	LD R, A	43
	16 bit-Ladebefehle	45
	LD dd, nn	47
	LD IX, nn	49
	LD IY, nn	51
	LD HL, (nn)	53
	LD dd, (nn)	55
	LD IX, (nn)	57
	LD IY, (nn)	59
	LD (nn), HL	61
	LD (nn), dd	63
	LD (nn), IX	65
	LD (nn), IY	67
	LD SP, HL	69

8 Hc 73



LD SP, IX	7	SBC A, r	149
LD SP, IY	7	SBC A, n	151
PUSH qq	7	SBC A, (HL)	153
PUSH IX	7	SBC A, (IX+d)	155
PUSH IY	7	SBC A, (IY+d)	157
POP qq	8	AND r	159
POP IX	8	AND n	161
POP IY	8	AND (HL)	163
EX DE, HL	8	AND (IX+d)	165
EX AF, AF'	8	AND (IY+d)	167
EXX	9	OR r	169
EX (SP), HL	9	OR n	171
EX (SP), IX	9	OR (HL)	173
EX (SP), IY	9	OR (IX+d)	175
Blocktransfer- und Suchbefehle	9	OR (IY+d)	177
LDI	10	XOR r	179
LDIR	10	XOR n	181
LDD	10	XOR (HL)	183
LDDR	10	XOR (IX+d)	185
CPI	10	XOR (IY+d)	187
CPIR	11	CP r	189
CPD	11	CP n	191
CPDR	11	CP (HL)	193
8 bit-Arithmetisch/Logische Befehle	11	CP (IX+d)	195
ADD A, r	11	CP (IY+d)	197
ADD A, n	12	INC r	199
ADD A, (HL)	12	INC (HL)	201
ADD A, (IX+d)	12	INC (IX+d)	203
ADD A, (IY+d)	12	INC (IY+d)	205
ADC A, r	12	DEC r	207
ADC A, n	13	DEC (HL)	209
ADC A, (HL)	13	DEC (IX+d)	211
ADC A, (IX+d)	13	DEC (IY+d)	213
ADC A, (IY+d)	13	DAA	215
SUB r	13	CPL	217
SUB n	14	NEG	219
SUB (HL)	14	16 bit-Arithmetisch/Logische Befehle	221
SUB (IX+d)	14	ADD HL, ss	223
SUB (IY+d)	14	ADC HL, ss	225



SBC HL, ss	227
ADD IX, pp	229
ADD IY, rr	231
INC ss	233
INC IX	235
INC IY	237
DEC ss	239
DEC IX	241
DEC IY	243
Rotations- und Schiebe-Befehle	245
RLCA	247
RLA	249
RRCA	251
RRA	253
RLC r	255
RLC (HL)	257
RLC (IX+d)	259
RLC (IY+d)	261
RL r	263
RL (HL)	265
RL (IX+d)	267
RL (IY+d)	269
RRC r	271
RRC (HL)	273
RRC (IX+d)	275
RRC (IY+d)	277
RR r	279
RR (HL)	281
RR (IX+d)	283
RR (IY+d)	285
SLA r	287
SLA (HL)	289
SLA (IX+d)	291
SLA (IY+d)	293
SRA r	295
SRA (HL)	297
SRA (IX+d)	299
SRA (IY+d)	301
SRL r	303

SRL (HL)	305
SRL (IX+d)	307
SRL (IY+d)	309
RLD	311
RRD	313
Befehle zur Einzelbit-Verarbeitung	315
BIT b, r	317
BIT b, (HL)	319
BIT b, (IX+d)	321
BIT b, (IY+d)	323
SET b, r	325
SET b, (HL)	327
SET b, (IX+d)	329
SET b, (IY+d)	331
RES b, r	333
RES b, (HL)	335
RES b, (IX+d)	337
RES b, (IY+d)	339
CPU-Steuerbefehle	341
CCF	343
SCF	345
NOP	347
HALT	349
DI	351
EI	353
IM 0	355
IM 1	357
IM 2	359
Sprungbefehle	361
JP nn	363
JP c, nn	365
JR e	367
JR C, e	369
JR NC, e	371
JR Z, e	373
JR NZ, e	375
JP (HL)	377
JP (IX)	379
JP (IY)	381



DJNZ e .....	383
CALL nn .....	385
CALL c, nn .....	387
RET .....	389
RET c .....	391
RETI .....	393
RETN .....	395
RST p .....	397
<b>Ein/Ausgabe-Befehle .....</b>	<b>399</b>
IN A, (n) .....	401
IN r, (C) .....	403
INI .....	405
INIR .....	407
IND .....	409
INDR .....	411
OUT (n), A .....	413
OUT (C), r .....	415
OUTI .....	417
OTIR .....	419
OUTD .....	421
OTDR .....	423

# 8 bit-Ladebefehle

Quellregister

	IMPLIED		REGISTER								REG INDIRECT			INDEXED		EXT. ADDR.	IMME
	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+d)	(nn)	n	
	ED 57	ED 5F	77	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n	3E n	
REGISTER	A												DD 46 d	FD 46 d		06 n	
	B												DD 4E d	FD 4E d		0E n	
	C												DD 56 d	FD 56 d		16 n	
	D												DD 5E d	FD 5E d		1E n	
	E												DD 66 d	FD 66 d		26 n	
	H												DD 6E d	FD 6E d		2E n	
	L												DD 7E d	FD 7E d		3E n	
REG INDIRECT	(HL)															3E n	
	(BC)																
	(DE)																
INDEXED	(IX+d)		DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d	
	(IY+d)		FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d	
EXT. ADDR	(nn)																
IMPLIED	I		ED 47														
	R		ED 4F														

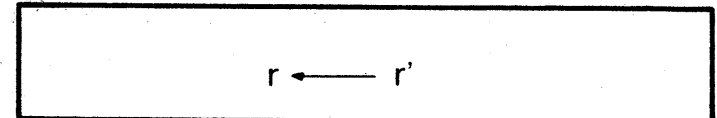
Zielregister



# LD r, r'

Datentransfer zwischen zwei Registern

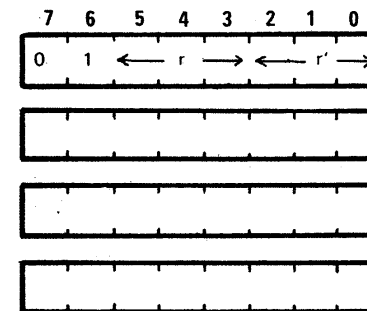
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	r,r'	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

<input type="text"/>	MSB
<input type="text"/>	2. Byte
<input type="text"/>	3. Byte
<input type="text"/>	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s/ 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

### Kommentar

Der Inhalt des Registers  $r'$  wird in das Register  $r$  kopiert. Zulässige Register  $r'$  und  $r$  sind A, B, C, D, E, H und L.  $r'$  und  $r$  werden im binären Maschinenprogramm jeweils durch folgende Bit-Kombinationen charakterisiert.

Register	A	B	C	D	E	H	L
$r'$ bzw. $r$	111	000	001	010	011	100	101

Beispiel: Register H enthalte den Wert 8AH und E den Wert 10H. Nach Ausführung von

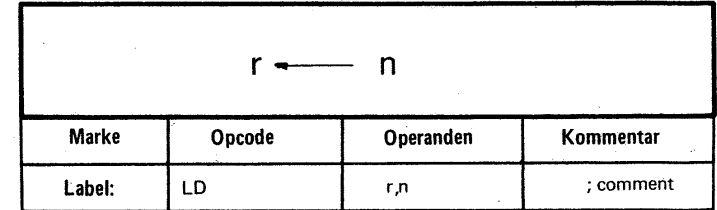
LD H,E

steht sowohl in H als auch in E der Wert 10H.

# LD r, n

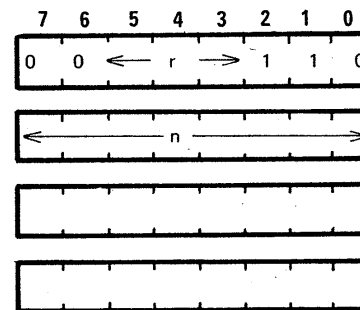
Laden eines Registers mit einer Konstanten

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

MSB
2. Byte
3. Byte
LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



### Kommentar

Die 8bit-Konstante n wird in das Register r geladen. Zulässige Register r sind A, B, C, D, h und L.  
r wird im binären Maschinenprogramm durch folgende Bit-Kombinationen charakterisiert:

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

Beispiel: Nach Ausführung von

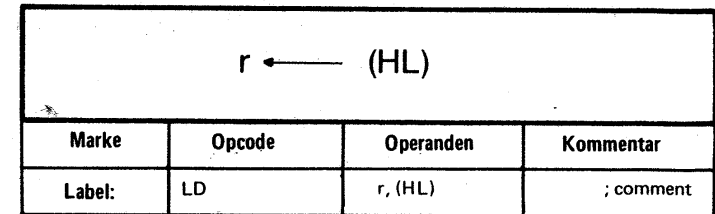
LD E,A5H

steht in E der Wert A5H, unabhängig davon, welchen Inhalt E vor Ausführung des Befehls hatte.

# LD r, [HL]

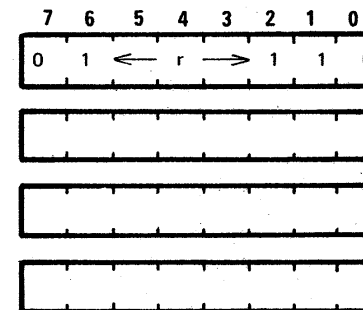
Laden eines Registers mit einer Speicherstelle

Was macht der Computer?

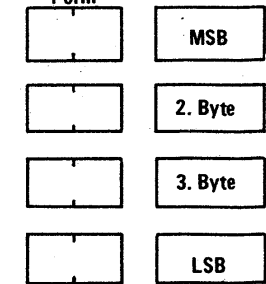


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

### Kommentar

Der 8bit umfassende Inhalt der Speicherstelle, die durch den Inhalt des Registerpaars HL adressiert ist, wird in das Register r kopiert.

Für r sind zulässig A, B, C, D, E, H und L.

r wird im binären Maschinenprogramm durch folgende Bit-Kombinationen charakterisiert:

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

Beispiel: Register H enthalte den Wert 75H, L enthalte A1H, während in der Speicherstelle mit der Adresse 75A1H der Wert 58H steht.

Nach Ausführung von

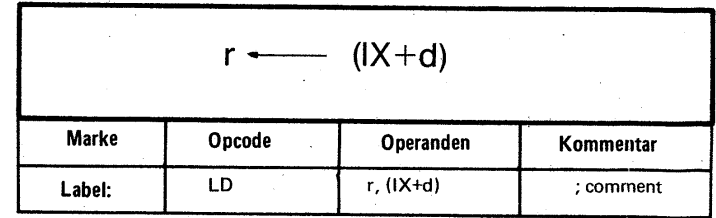
LD C, (HL)

steht in Register C der Wert 58H, unabhängig davon, welchen Inhalt C vor Ausführung des Befehls hatte.

# LD r, [IX+d]

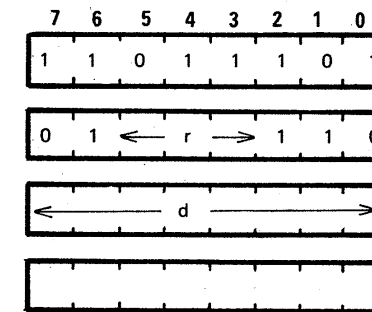
Laden eines Registers mit einer indiziert adressierten Speicherstelle

Was macht der Computer?

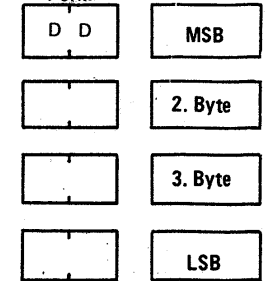


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

### Kommentar

Der Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IX und der 8bit-Distanz \*d adressiert ist, wird in das Register r kopiert.

Negative Distanzangaben werden vom Assembler in Form des Zweierkomplements dargestellt.

Für Register r sind A, B, C, D, E, H und L zulässig. Im binären Maschinencode werden diese wie folgt codiert:

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

Beispiel: Das 16bit-Indexregister enthalte den Wert 25AFH.  
Die Ausführung von

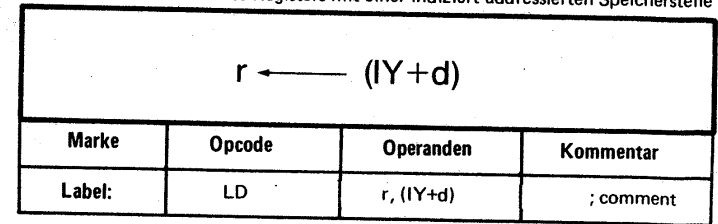
LD B, (IX + 19H)

bewirkt die Berechnung der anzusprechenden Speicherstelle aus Index plus Distanz (also 25AFH + 19H = 25C8H). Falls diese Speicherstelle 25C8H den Wert 39H enthält, steht nach Ausführung des Befehls auch im Register B der Wert 39H, unabhängig davon, welchen Inhalt B vorher hatte.

\*Bemerkung: Darstellung von d erfolgt in Form des Zweierkomplements.

# LD r, [IX+d]

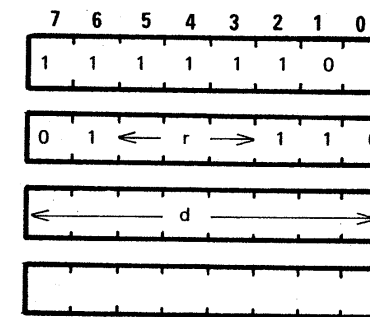
Laden eines Registers mit einer indiziert adressierten Speicherstelle



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu\text{s}$ / 7.6 $\mu\text{s}$	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



### Kommentar

Der Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IY und der 8bit-Distanz \*d adressiert ist, wird in das Register r kopiert.

Negative Distanzangaben werden vom Assembler in Form des Zweierkomplements dargestellt.

Für Register r sind A, B, C, D, E, H und L zulässig. Im binären Maschinencode werden diese wie folgt codiert:

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

Beispiel: Das 16bit-Indexregister enthalte den Wert 25AFH.  
Die Ausführung von

LD B, (IY + 19H)

bewirkt die Berechnung der anzusprechenden Speicherstelle aus Index plus Distanz (also 25AFH + 19H = 25C8H). Falls diese Speicherstelle 25C8H den Wert 39H enthält, steht nach Ausführung des Befehls auch im Register B der Wert 39H, unabhängig davon, welchen Inhalt B vorher hatte.

\* Bermerkung: Darstellung von d erfolgt in Form des Zweierkomplements.

# LD [HL], r

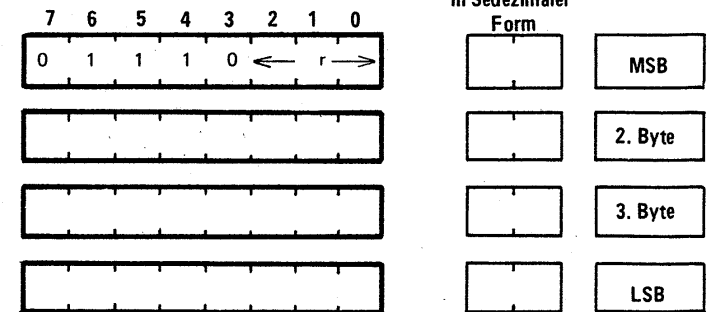
Laden einer Speicherstelle mit einem Registerinhalt

(HL) ← r			
Marke	Opcode	Operanden	Kommentar
Label:	LD	(HL),r	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Der Inhalt des Register r wird in die Speicherstelle kopiert, die durch den Inhalt des HL-Registerpaares adressiert ist. Dabei ist r eines der Register A, B, C, D, E, H oder L, die im Maschinencode wie folgt codiert werden:

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

Beispiel: Im Register H stehe momentan 21H, im Register L 46H. Der momentan in B abgelegte Wert sei 29H. Nach Ausführung des Befehls

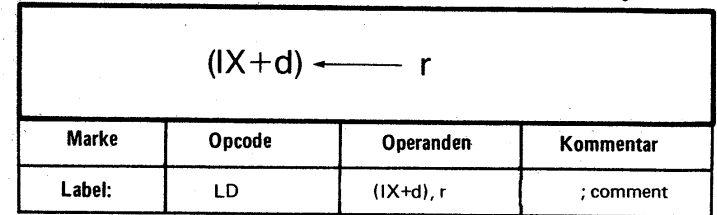
LD (HL), B

enthält dann sowohl das Register B, als auch die Speicherstelle 2146H jeweils den Wert 29H, unabhängig vom bisherigen Inhalt von 2146H.

# LD [IX+d], r

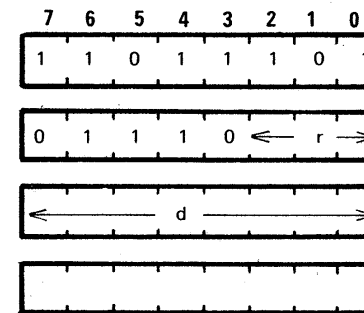
Laden einer indiziert adressierten Speicherstelle mit einem Registerinhalt

Was macht der Computer?

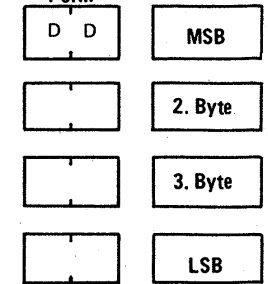


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s/ 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Der Inhalt des Registers r wird in die Speicherstelle kopiert, die durch die Summe aus dem Inhalt des 16 bit-Indexregisters IX und der 8 bit-Distanz d adressiert ist. Dabei ist r eines der Register A, B, C, D, E, H oder L, die im Maschinencode wie folgt codiert werden:

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

Beispiel: Im Doppelregister IX stehe momentan der Wert 2146H, der momentan in B abgelegte Wert sei 29H.  
Nach Ausführung des Befehls

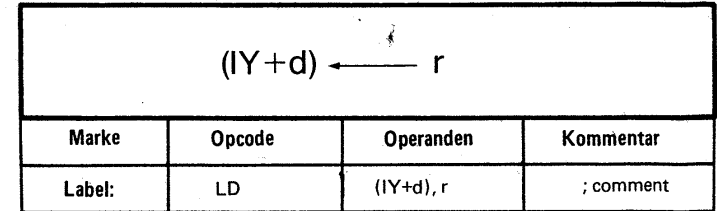
LD (IX+1),B

enthält dann sowohl das Register B, als auch die Speicherstelle 2147H jeweils den Wert 29H, unabhängig vom bisherigen Inhalt von 2147H.

# LD [IX+d], r

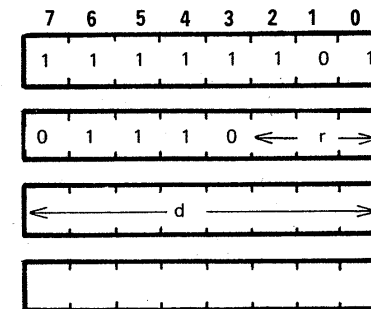
Laden einer indiziert adressierten Speicherstelle mit einem Registerinhalt

Was macht der Computer?

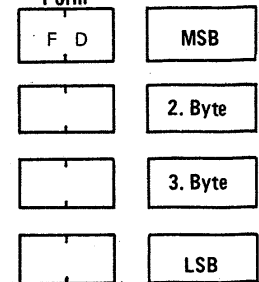


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s/ 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



### Kommentar

Der Inhalt des Registers r wird in die Speicherstelle kopiert, die durch die Summe aus dem Inhalt des 16 bit-Indexregisters IY und der 8 bit-distanz d adressiert ist. Dabei ist r eines der Register A, B, C, D, E, H oder L, die im Maschinencode wie folgt codiert werden:

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

Beispiel: Im Doppelregister IY stehe momentan der Wert 2146H, der momentan in B abgelegte Wert sei 29H.  
Nach Ausführung des Befehls

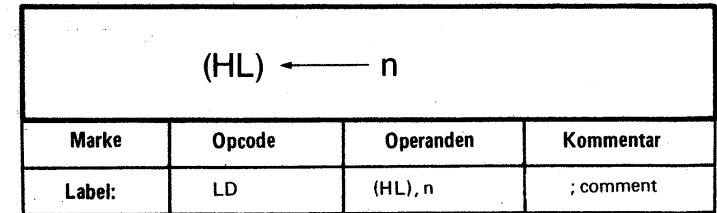
LD (IY+1),B

enthält dann sowohl das Register B, als auch die Speicherstelle 2147H jeweils den Wert 29H, unabhängig vom bisherigen Inhalt von 2147H.

# LD [HL], n

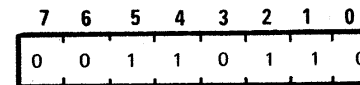
Laden einer Speicherstelle mit einer Konstanten

Was macht der Computer?

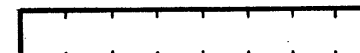
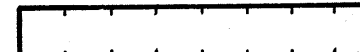
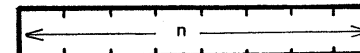


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



3 6	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Die 8 bit-Konstante n wird in die Speicherstelle geladen, die durch den Inhalt des Registerpaars HL adressiert ist.

Beispiel: H enthalte den Wert 44H und L den Wert 43H.  
Der Befehl

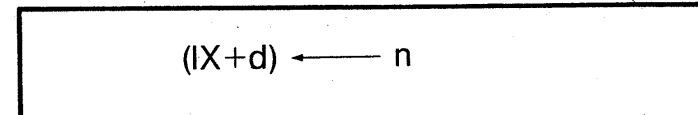
LD (HL), 28H

würde dann ein Laden der Speicherstelle 4443H mit dem Wert 28H veranlassen, unabhängig vom bisherigen Inhalt dieser Speicherstelle.

# LD [IX+d], n

Laden einer indiziert adressierten Speicherstelle mit einer Konstanten

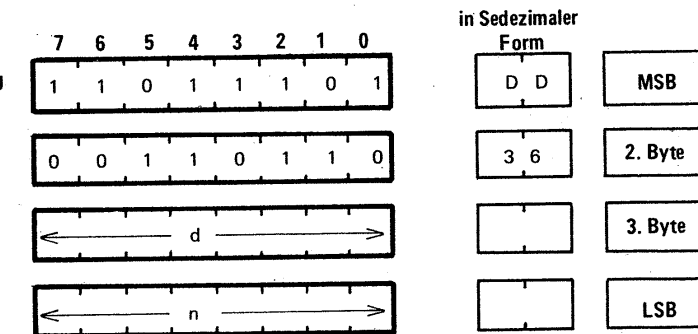
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	(IX+d), n	; comment

Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Die 8bit-Konstante n wird in die Speicherstelle geladen, die durch die Summe aus Indexregisterinhalt IX und Distanz d adressiert ist. Negative Distanzangaben werden vom Assembler in Form des Zweierkomplements dargestellt.

Beispiel: Das Indexregister IX enthalte den 16bit-Wert 219AH.  
Die Anweisung

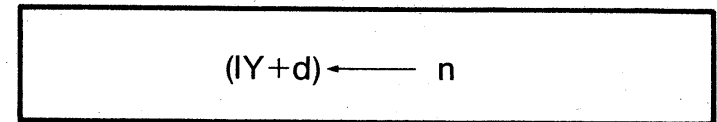
LD (IX+5H), 5AH

würde zunächst die anzusprechende Adresse durch Addition des Indexregisterinhalts mit der Distanz (219AH+5H = 219FH) ermitteln und dann die Konstante 5AH in die Speicherstelle 219FH laden. Der bisherige Inhalt dieser Speicherstelle ist dabei unerheblich und wird überschrieben.

# LD [IX+d], n

Laden einer indiziert adressierten Speicherstelle mit einer Konstanten

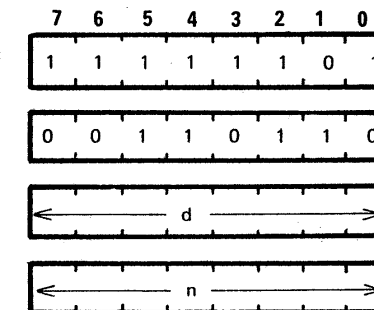
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	(IX+d), n	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

F D	MSB
3 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinentzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Die 8bit-Konstante n wird in die Speicherstelle geladen, die durch die Summe aus Indexregisterinhalt IY und Distanz d adressiert ist. Negative Distanzangaben werden vom Assembler in Form des Zweierkomplements dargestellt.

Beispiel: Das Indexregister IY enthalte den 16bit-Wert 219AH.  
Die Anweisung

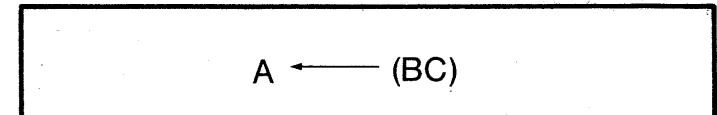
LD (IY+5H), 5AH

würde zunächst die anzusprechende Adresse durch Addition des Indexregisterinhalts mit der Distanz (219AH+5H = 219FH) ermitteln und dann die Konstante 5AH in die Speicherstelle 219FH laden. Der bisherige Inhalt dieser Speicherstelle ist dabei unerheblich und wird überschrieben.

# LD A, [BC]

Laden des Akkus mit einer Speicherstelle

Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	A, (BC)	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

in Sedezimaler Form

0 A	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der Speicherstelle, die durch den Inhalt des Registerpaars BC adressiert ist, wird in das A-Register kopiert.

Beispiel: Das Register B enthalte den Wert 47H, C den Wert 11H und in der Speicherstelle mit der Adresse 4711H stehe der Wert 12H.  
Der Befehl

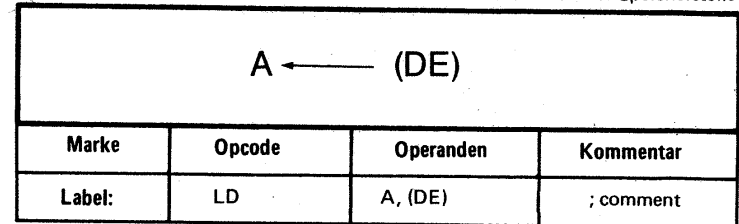
LD A, (BC)

bewirkt dann, daß nach seiner Ausführung in Register A der Wert 12H unabhängig von seinem vorhergehenden Inhalt steht.

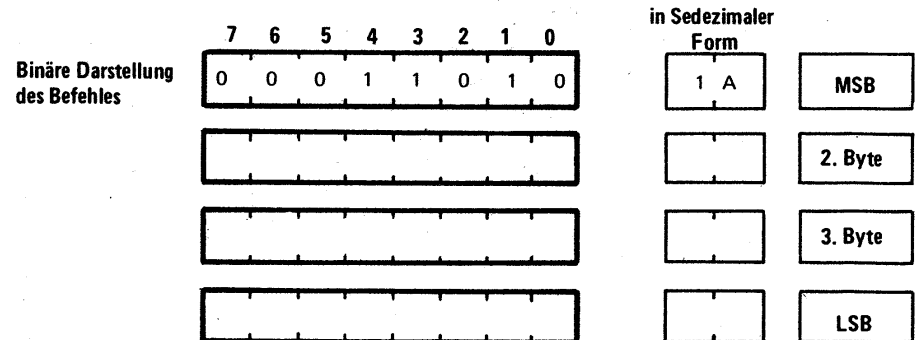
# LD A, [DE]

Laden des Akkus mit einer Speicherstelle

Was macht der Computer?



Befehlsformat in Assembler



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflußt

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Der Inhalt der Speicherstelle, die durch den Inhalt des Registerpaars DE adressiert ist, wird in das A-Register kopiert.

Beispiel: Das Register D enthalte den Wert 47H, E den Wert 11H und in der Speicherstelle mit der Adresse 4711H stehe der Wert 12H.  
Der Befehl

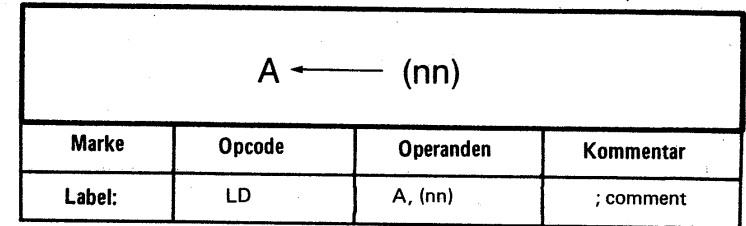
LD A, (DE)

bewirkt dann, daß nach seiner Ausführung in Register A der Wert 12H unabhängig von seinem vorhergehenden Inhalt steht.

# LD A, [nn]

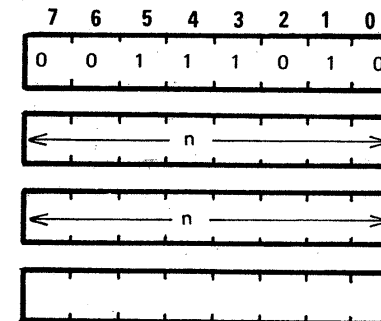
Laden des Akkus mit einer Speicherstelle

Was macht der Computer?

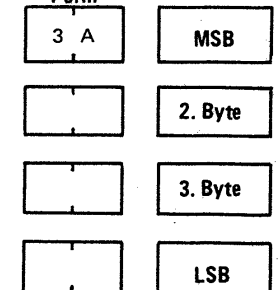


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.25 $\mu$ s / 5.2 $\mu$ s	4	13 (4,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch die 16bit-Konstante nn adressiert ist, wird in den Akkumulator A kopiert. nn kann selbstverständlich auch als symbolischer Name für ein 16bit-Wort angegeben werden.

Das auf den Opcode direkt folgende Byte stellt dabei den niederwertigeren Teil der Adresse dar.

Beispiel: Hat die Speicherstelle mit der Adresse 8832H den Inhalt 04H, so bewirkt der Befehl

LD A,(8832H),

daß nach seiner Ausführung sowohl in der Speicherstelle 8832H, als auch dem Akkumulator der Wert 04H steht, unabhängig von dessen bisherigem Inhalt.

# LD [BC], A

Laden einer Speicherstelle mit Akkuinhalt

Was macht der Computer?

(BC) ← A

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	(BC),A	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

in Sedezimaler Form

0 2	MSB
-----	-----

--	--	--	--	--	--	--	--

	2. Byte
--	---------

--	--	--	--	--	--	--	--

	3. Byte
--	---------

--	--	--	--	--	--	--	--

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der 8bit-Inhalt des Akkumulators H wird in die Speicherstelle kopiert, die durch den Inhalt des Registerpaars BC adressiert ist.

Beispiel: Register A habe den Inhalt 7AH, in B stehe der Wert 12H und in C der Wert 13H.  
Nach Ausführung von

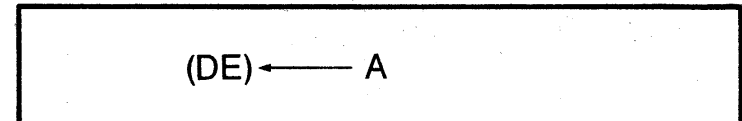
LD (BC), A

steht der Wert 7AH sowohl im Register A, als auch in der Speicherstelle 1213H, unabhängig vom bisherigen Inhalt dieser Speicherstelle.

# LD [DE], A

Laden einer Speicherstelle mit Akkuinhalt

Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	(DE), A	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	1	0	0	1	0

in Sedezimaler Form

1 2	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der 8bit-Inhalt des Akkumulators H wird in die Speicherstelle kopiert, die durch den Inhalt des Registerpaars DE adressiert ist.

Beispiel: Register A haben den Inhalt 7AH, in D stehe der Wert 12H und in E der Wert 13H.  
Nach Ausführung von

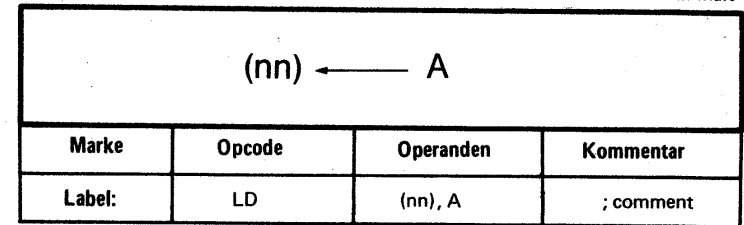
LD (DE), A

steht der Wert 7AH sowohl im Register A, als auch in der Speicherstelle 1213H, unabhängig vom bisherigen Inhalt dieser Speicherstelle.

# LD [nn], A

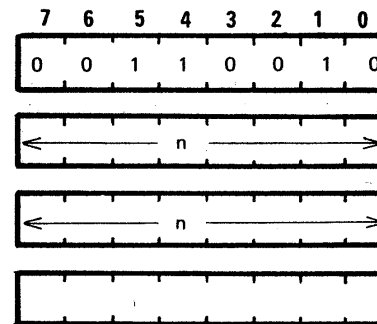
Laden einer Speicherstelle mit Akkuinhalt

Was macht der Computer?

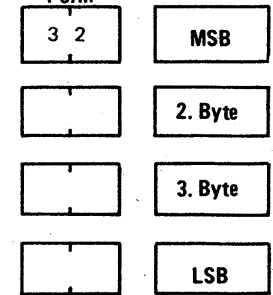


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.25 $\mu$ s / 5.2 $\mu$ s	4	13 (4,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der 8bit-Inhalt des Akkumulators A wird in die Speicherstelle kopiert, die durch die 16bit-Konstante nn adressiert ist.

nn kann selbstverständlich auch als symbolischer Name für ein 16bit-Wort angegeben werden.

Das auf den Opcode direkt folgende Byte stellt dabei den niederwertigeren Teil der Adresse dar.

Beispiel: Der Akkumulator enthalte den Wert D7H.  
Nach Ausführung von

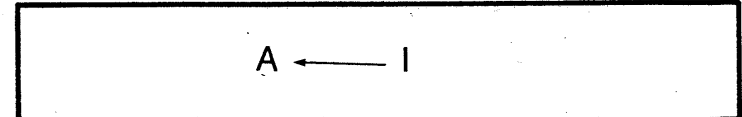
LD (3141H), A

steht der Wert D7H sowohl im Akkumulator, als auch in der Speicherstelle 3141H, unabhängig von deren bisherigem Inhalt.

# LD A, I

Laden des Akkus vom Interrupt-Register

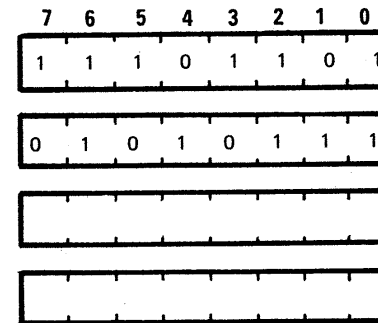
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	A,I	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

E D	MSB
5 7	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.25 $\mu$ s / 3.6 $\mu$ s	2	9 (4,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, falls Inhalt von I negativ, sonst rückgesetzt
Z	Gesetzt, falls Inhalt von I gleich Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Enthält den Inhalt von IFF2
N	Rückgesetzt
C	Nicht beeinflusst



## Kommentar

Der Inhalt des Interruptvektor-Registers I wird in den Akkumulator A kopiert.

Beispiel: Das Interruptvektor-Register enthalte den Wert 4AH.  
Nach Ausführung von

LD A,I

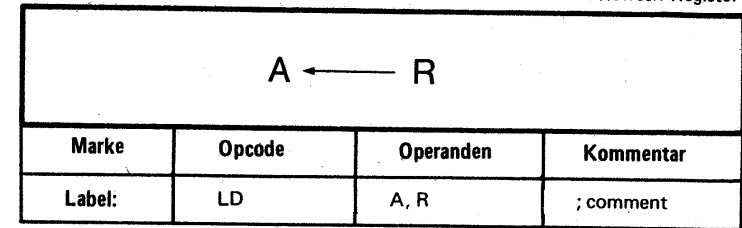
enthalten sowohl Register I, als auch A den Wert 4AH,  
unabhängig vom bisherigen Akkuinhalt.

Bemerkung: Falls während der Ausführung dieses Befehls eine Interrupt-Anforderung auftritt, wird das Paritätsbedingungsbit rückgesetzt.

# LD A, R

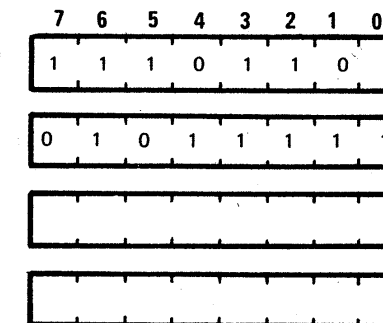
Laden des Akkus vom Refresh-Register

Was macht der Computer?

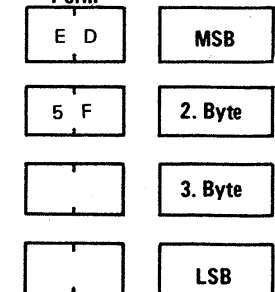


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.25 $\mu$ s / 3.6 $\mu$ s	2	9 (4,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, falls Inhalt von R negativ, sonst rückgesetzt
Z	Gesetzt, falls Inhalt von R gleich Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Enthält den Inhalt von IFF2
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des Refresh-Registers R wird in den Akkumulator A kopiert.

Beispiel: Das Refresh-Register R enthalte den Wert 4AH.  
Nach Ausführung von

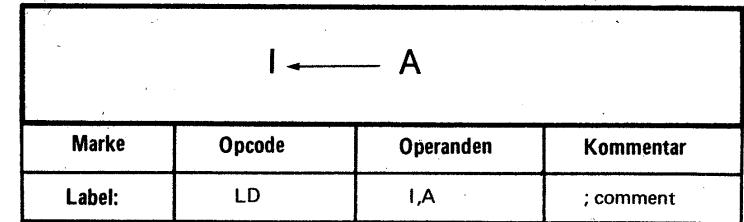
LD A,R

enthalten sowohl R als auch A den Wert 4AH, unabhängig vom bisherigen Inhalt von A.

# LD I, A

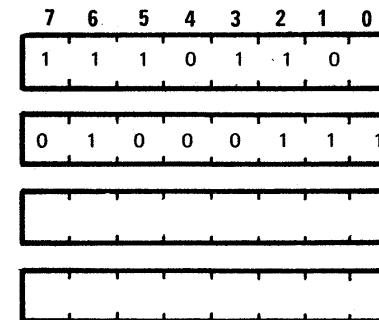
Laden des Interrupt-Registers vom Akku

Was macht der Computer?

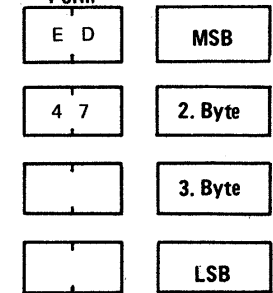


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.25 $\mu\text{s}$ / 3.6 $\mu\text{s}$	2	9 (4,5)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Der 8bit-Inhalt des Akkumulators A wird in das Interruptvektor-Register I kopiert.

Beispiel: Der Akkumulator haben den Inhalt 80H.  
Nach Ausführung von

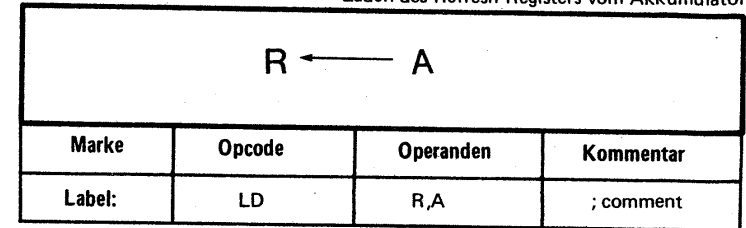
LD I,A

enthalten sowohl A als auch I den Wert 80H, unabhängig vom bisherigen Inhalt von I.

# LD R, A

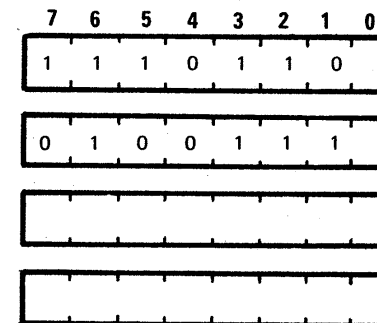
Laden des Refresh-Registers vom Akkumulator

Was macht der Computer?

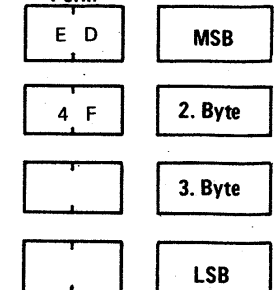


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.25 $\mu$ s / 3.6 $\mu$ s	2	9 (4,5)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Der Inhalt des Akkumulators A wird in das Refreshregister R kopiert.

Beispiel: Der Akkumulator haben den Inhalt 04H.  
Nach Ausführung von

LD R,A

enthalten sowohl A als auch R den Wert 04H, unabhängig vom bisherigen Inhalt von R.

# 16 bit-Ladebefehle

Quellregister

		REGISTER						IMM. EXT.	EXT. ADDR.	REG. INDIR.	
		AF	BC	DE	HL	SP	IX	IY	nn	(nn)	(SP)
Zielregister	REGISTER	AF									E1
	BC							01 n n	ED 4B n n	C1	
	DE							11 n n	ED 5B n n	D1	
	HL							21 n n	2A n n	E1	
	SP				FD n	DD F9	FD F9	31 n n	ED 7B n n		
	IX							DD 21 n n	DD 2A n n	DD E1	
	IY							FD 21 n n	FD 2A n n	FD E1	
EXT. ADDR.	(nn)		ED 43 n n	ED 53 n n	FD n n	ED 73 n n	DD 22 n n	FD 22 n n			
REG. IND.	(SP)	FD n	CD n	DD n	ED n		DD E5	FD E5			

PUSH INSTRUCTIONS →      ← POP INSTRUCTIONS

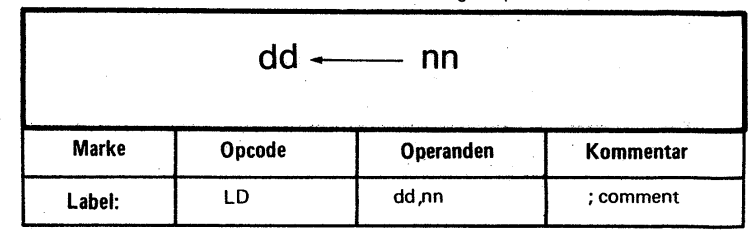
Bemerkung: PUSH- und POP-Befehle beeinflussen auch den Stackpointer entsprechend.

		IMPLIED ADDRESSING				
		AF	BC, DE & HL	HL	IX	IY
IMPLIED	AF	08				
	BC, DE & HL		D9			
	DE			E3		
REG. INDIR.	(SP)			E3	DD E3	FD E3

# LD dd, nn

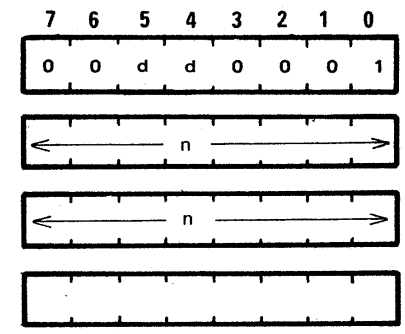
Laden eines Registerpaares mit einer Konstanten

Was macht der Computer?

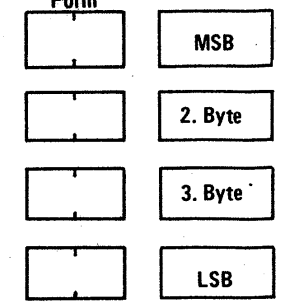


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Die 16bit-Konstante nn wird in das Registerpaar dd geladen, wobei für dd eines der Registerpaare BC, DE, HL oder SP zulässig ist. Der entsprechende Maschinencode ist

Registerpaar	BC	DE	HL	SP
dd	00	01	10	11

Das unmittelbar auf den Opcode folgende Byte n ist das niederwertigere Byte des 16bit-Wortes nn.

Beispiel: Nach Ausführung des Befehls

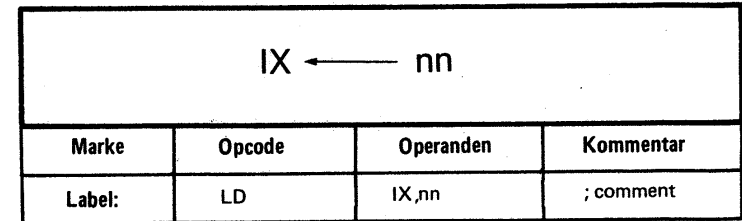
LD HL, 5000H

ist der Inhalt des Registers H = 50H und der von L = 0H, unabhängig von den bisherigen Inhalten dieser Register.

# LD IX, nn

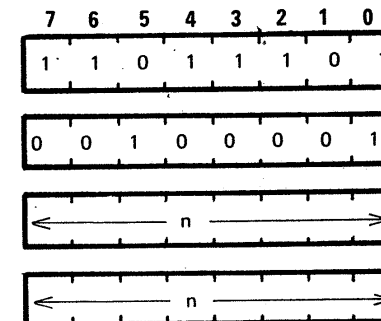
Laden eines Indexregisters mit einer Konstanten

Was macht der Computer?

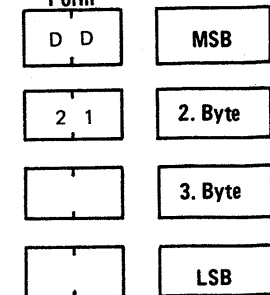


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu$ s / 5.6 $\mu$ s	4	14 (4,4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



### Kommentar

Die 16bit-Konstante nn wird in das Indexregister IX geladen. Das auf den Opcode unmittelbar folgende Byte n ist das niederwertige Byte des 16bit-Wortes nn.

Beispiel: Nach Ausführung des Befehls

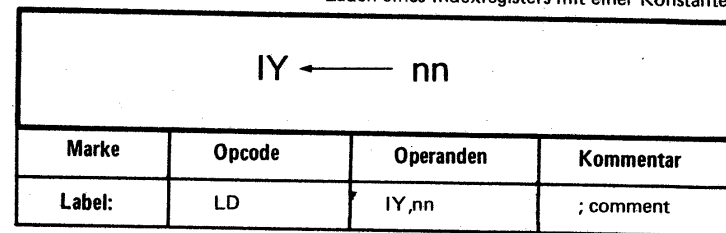
LD IX, 45A2H

enthält IX den Wert 45A2H, unabhängig von seinem bisherigen Inhalt.

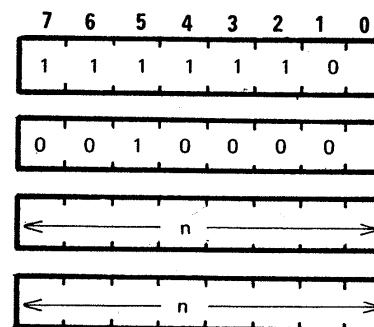
# LD IX, nn

Laden eines Indexregisters mit einer Konstanten

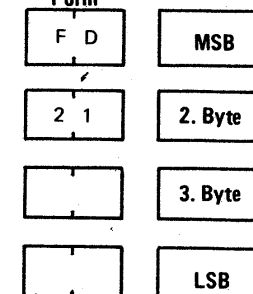
Was macht der Computer?



Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu$ s / 5.6 $\mu$ s	4	14 (4,4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Die 16bit-Konstante nn wird in das Indexregister IY geladen. Das auf den Opcode unmittelbar folgende Byte n ist das niederwertige Byte des 16bit-Wortes nn.

Beispiel: Nach Ausführung des Befehls

LD IY, 45A2H

enthält IY den Wert 45A2H, unabhängig von seinem bisherigen Inhalt.

# LD HL, [nn]

Laden eines Registerpaars mit zwei Speicherstellen

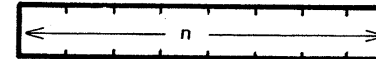
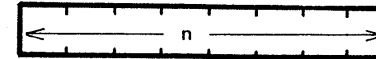
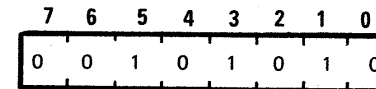
H ← (nn+1), L ← (nn)

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	HL,(nn)	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

2 A	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6,4 $\mu$ s	5	16 (4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der Speicherstelle, die durch das 16bit-Wort nn adressiert ist, wird ins Register L kopiert.

Der Inhalt der Speicherstelle, die durch nn+1 adressiert ist, wird ins Register H kopiert.

Das auf den Opcode unmittelbar folgende Byte n ist das niederwertige Byte des 16bit-Wortes nn.

Beispiel: Die Speicherstelle mit der Adresse 4545H enthält den Wert 37H, Speicherstelle 4546H den Wert A1H.  
Nach Ausführung von

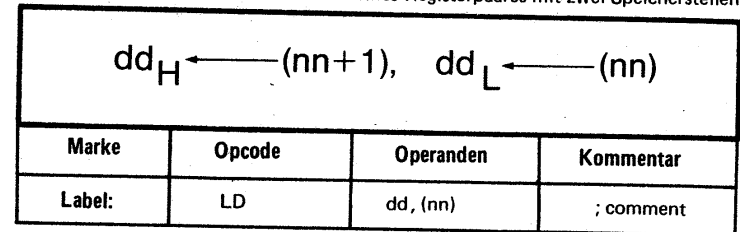
LD HL (4545H)

enthält Register H den Wert A1H und Register L den Wert 37H, unabhängig vom bisherigen Inhalt dieser Register.

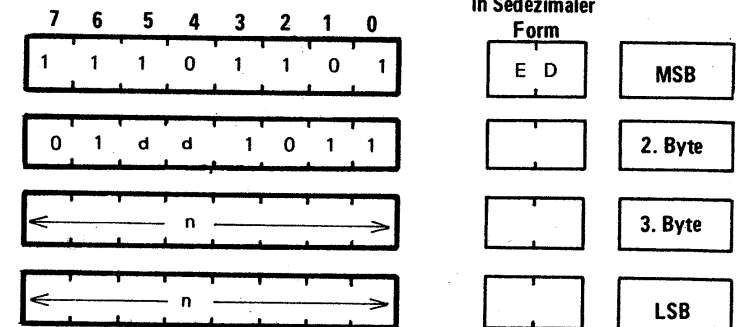
# LD dd, [nn]

Laden eines Registerpaares mit zwei Speicherstellen

Was macht der Computer?



Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu\text{s}$ / 8 $\mu\text{s}$	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der durch die 16bit-Kontante nn adressierten Speicherstelle wird in die niederwertige Hälfte des Registerpaares dd (also in C, E, L oder SP<sub>L</sub>) kopiert. Der Inhalt der durch nn+1 adressierten Speicherstelle wird in die höherwertige Hälfte des Registerpaares dd (also in B, D, H oder SP<sub>H</sub>) kopiert.

Als Registerpaare sind zulässig BD, DE, HL und das Doppelregister SP; angesprochen werden sie durch folgende Bitkombination im Maschinencode:

16bit-Register	BC	DE	HL	SP
dd	00	01	10	11

Das auf den Opcode unmittelbar folgende Byte n ist das niederwertigere Byte des 16bit-Wortes nn.

Beispiel: Die durch 2130H adressierte Speicherstelle enthalte den Wert 65H und Speicherstelle 2131H den Wert 78H.  
Nach Ausführung des Befehls

LD BC, (2130H)

steht in B der Wert 78H und in C der Wert 65H, unabhängig vom bisherigen Inhalt dieser Register.

# LD IX, [nn]

Laden eines Indexregisters mit zwei Speicherstellen

Was macht der Computer?

IX<sub>H</sub> ← (nn+1), IX<sub>L</sub> ← (nn)

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	IX, (nn)	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	0	1	1	0	1	1

in Sedezimaler Form

D D	MSB
-----	-----

0	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

2 A	2. Byte
-----	---------

← n →
-------

	3. Byte
--	---------

← n →
-------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu$ s / 8 $\mu$ s	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der durch die 16bit-Konstante nn adressierte Speicherstelle wird in die niederwertige Hälfte des Indexregisters IX kopiert. Der Inhalt der durch nn+1 adressierten Speicherstelle wird in die höherwertige Hälfte von IX kopiert. Das unmittelbar auf den Opcode folgende Byte n ist der niederwertige Teil des 16bit-Wortes nn.

Beispiel: Die durch 6666H adressierte Speicherstelle enthalte den Wert 92H, Speicherstelle 6667H den Wert DAH.  
Nach Ausführung des Befehls

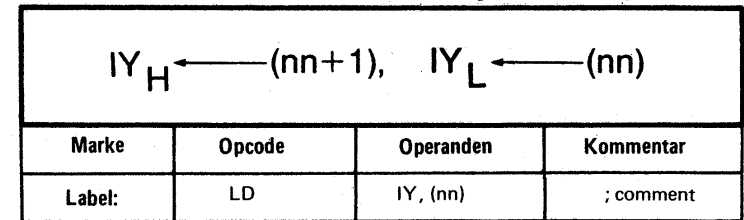
LD IX, (6666H)

enthält IX den Wert DA92H, unabhängig von seinem bisherigen Inhalt.

# LD IX, [nn]

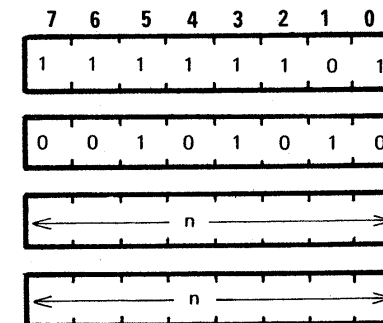
Laden eines Indexregisters mit zwei Speicherstellen

Was macht der Computer?

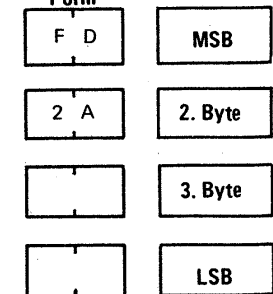


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu\text{s}$ / 8 $\mu\text{s}$	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Der Inhalt der durch die 16bit-Konstante nn adressierte Speicherstelle wird in die niederwertige Hälfte des Indexregisters IY kopiert. Der Inhalt der durch nn+1 adressierten Speicherstelle wird in die höherwertige Hälfte von IY kopiert. Das unmittelbar auf den Opcode folgende Byte n ist der niederwertige Teil des 16bit-Wortes nn.

Beispiel: Die durch 6666H adressierte Speicherstelle enthalte den Wert 92H, Speicherstelle 6667H den Wert DAH.  
Nach Ausführung des Befehls

LD IY, (6666H)

enthält IY den Wert DA92H, unabhängig von seinem bisherigen Inhalt.

# LD [nn], HL

Laden zweier Speicherstellen von einem Registerpaar aus

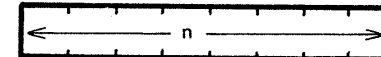
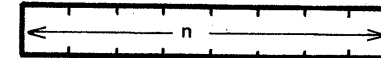
(nn+1) ← H, (nn) ← L			
Marke	Opcode	Operanden	Kommentar
Label:	LD	(nn), HL	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
0	0	1	0	0	0	1	0



in Sedezimaler Form

2 2	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu\text{s}$ / 6.4 $\mu\text{s}$	5	16 (4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Der Inhalt von Register L wird in die durch die 16bit-Konstante nn adressierte Speicherstelle kopiert. Der Inhalt von Register H wird in die Speicherstelle nn+1 kopiert.

Das unmittelbar auf den Opcode folgende Byte n ist das niederwertige Byte des 16bit-Wortes nn.

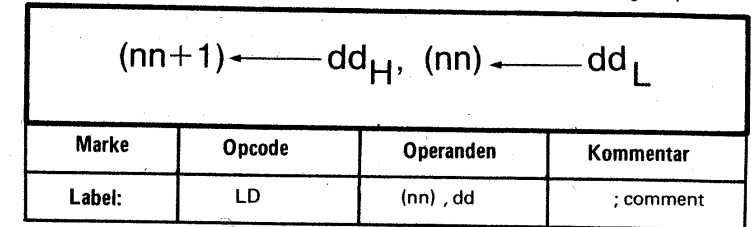
Beispiel: Der Inhalt von Register H sei 48H, der von L 3AH. Nach Ausführung des Befehls

LD (B229H), HL

enthält die durch B229H adressierte Speicherstelle den Wert 3AH und die Adresse B22AH den Wert 48H, unabhängig vom bisherigen Inhalt der beiden Speicherstellen.

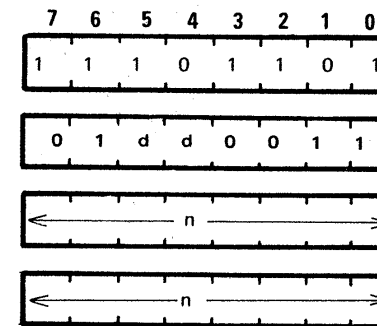
# LD [nn], dd

Laden zweier Speicherstellen von einem Registerpaar aus

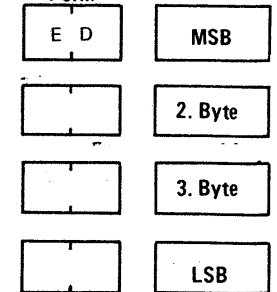


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu\text{s}$ / 8 $\mu\text{s}$	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

### Kommentar

Der Inhalt des niederwertigen Teils des 16bit-Registers dd wird in die durch die 16bit-Konstante nn adressierte Speicherstelle kopiert. Der Inhalt des höherwertigen Teils von dd wird in die Speicherstelle nn+1 kopiert. Als 16bit-Register dd sind zulässig die Registerpaare BC, DE, HL und das Doppelregister SP; sie werden im Objekt-Code durch folgende bit-Kombinationen angesprochen:

16bit-Register	BC	DE	HL	SP
dd	00	01	10	11

Das auf den Opcode unmittelbar folgende Byte n ist der niederwertige Teil des 16bit-Wortes nn.

Beispiel: Register B enthalte den Wert 46H, C den Wert 44H.  
Nach Ausführung des Befehls

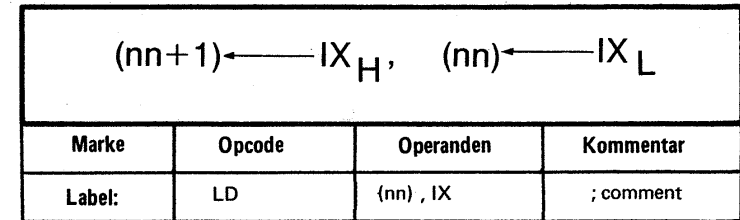
LD (1000H), BC

steht in der durch das 16bit-Wort 1000H adressierten Speicherstelle der Wert 44H und in Speicherstelle 1001H der Wert 46H.

# LD [nn], IX

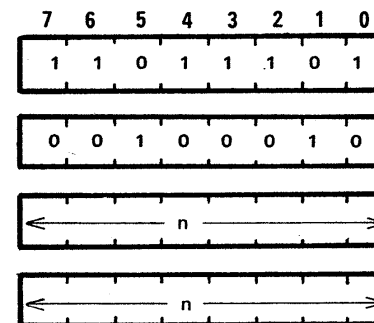
Laden zweier Speicherstellen von einem Indexregister aus

Was macht der Computer?

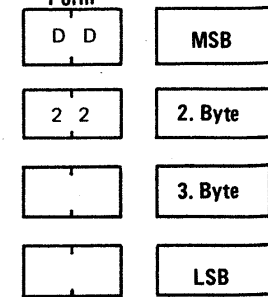


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu$ s / 8 $\mu$ s	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Das niederwertige Byte des Indexregisters IX wird in die durch das 16bit-Wort nn adressierte Speicherstelle kopiert. Das höherwertige Byte von IX wird in die Speicherstelle nn+1 kopiert. Das unmittelbar auf den Opcode folgende Byte n ist der niederwertige Teil des 16bit-Wortes nn.

Beispiel: Das Indexregister IX enthalte den Wert 5A30A.  
Nach Ausführung des Befehls

LD (4392H), IX

enthält die durch 4392H adressierte Speicherstelle den Wert 30H und die Speicherstelle 4393H den Wert 5AH, unabhängig vom bisherigen Inhalt dieser beiden Speicherstellen.

# LD [nn], IX

Laden zweier Speicherstellen von einem Indexregister aus

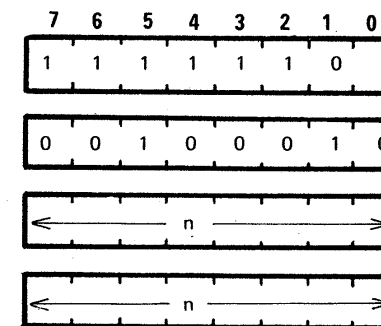
Was macht der Computer?

(nn+1) ← IY<sub>H</sub>, (nn) ← IY<sub>L</sub>

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	(nn), IY	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

F D	MSB
2 2	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu$ s / 8 $\mu$ s	6	20 (4,4,3,3,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Das niederwertige Byte des Indexregisters IY wird in die durch das 16bit-Wort nn adressierte Speicherstelle kopiert. Das höherwertige Byte von IY wird in die Speicherstelle nn+1 kopiert. Das unmittelbar auf den Opcode folgende Byte n ist der niederwertige Teil des 16bit-Wortes nn.

Beispiel: Das Indexregister IY enthalte den Wert 5A30A.  
Nach Ausführung des Befehls

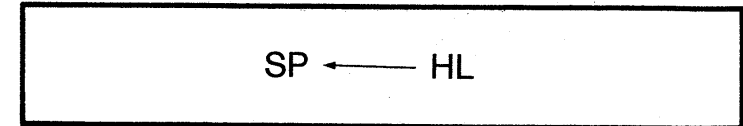
LD (4392H), IY

enthält die durch 4392H adressierte Speicherstelle den Wert 30H und die Speicherstelle 4393H den Wert 5AH, unabhängig vom bisherigen Inhalt dieser beiden Speicherstellen.

# LD SP, HL

Laden des Stackpointers von einem Registerpaar aus

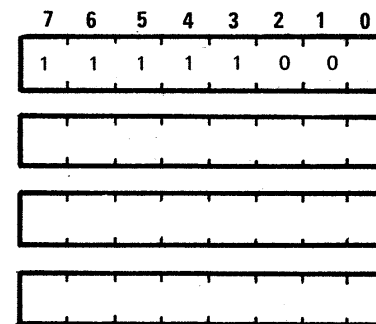
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LD	SP,HL	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

F 9	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.5 $\mu$ s/ 2.4 $\mu$ s	1	6

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Der Inhalt des Registerpaars HL wird in das Stackpointer-Doppelregister SP kopiert.

Beispiel: Das Register H enthalte den Wert 44H, L den Wert 2EH. Nach Ausführung des Befehls

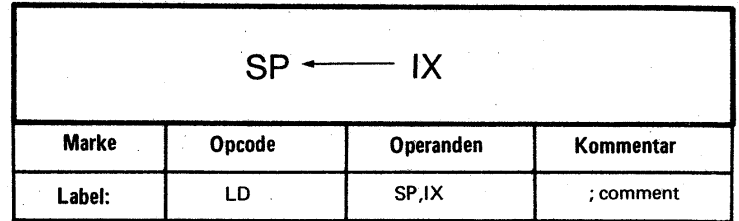
LD SP,HL

enthält der Stackpointer den Wert 442EH, unabhängig von seinem bisherigen Inhalt.

# LD SP, IX

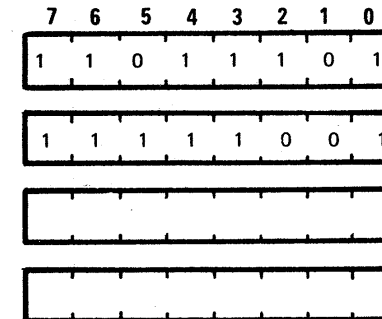
Laden des Stackpointers von einem Indexregister aus

Was macht der Computer?

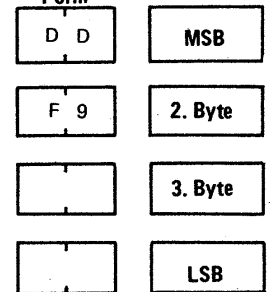


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	2	10 (4,6)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Der 16bit-Inhalt des Doppelregisters IX wird in das 16bit-Stackpointerregister SP kopiert.

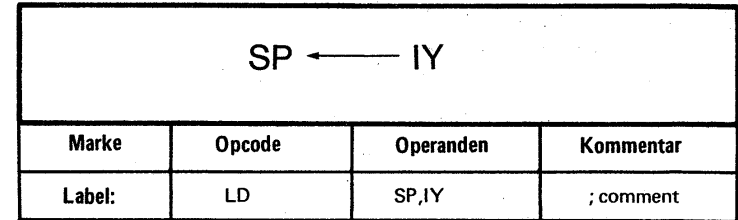
Beispiel: Der Inhalt IX sei 98DAH.  
Nach Ausführung des Befehls

LD SP,IX

ist der Inhalt von IX und von SP 98DAH, unabhängig vom bisherigen Inhalt von SP.

# LD SP, IY

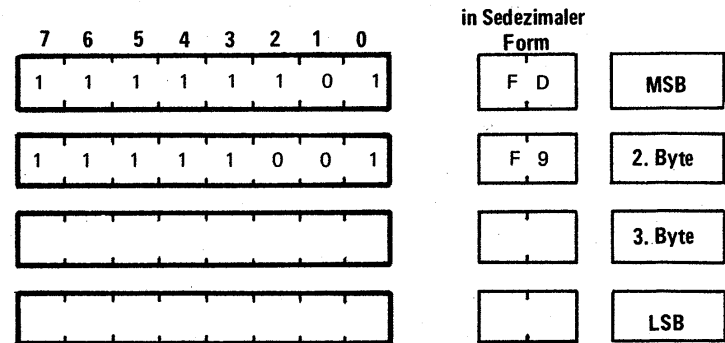
Laden des Stackpointers von einem Indexregister aus



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu\text{s}$ / 4 $\mu\text{s}$	2	10 (4,6)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der 16bit-Inhalt des Doppelregisters IY wird in das 16bit-Stackpointerregister SP kopiert.

Beispiel: Der Inhalt von IY sei 98DAH.  
Nach Ausführung des Befehls

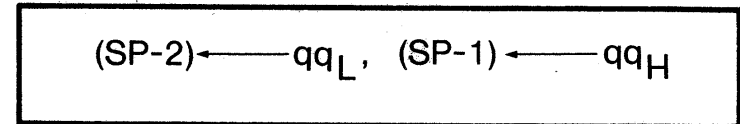
LD SP,IY

ist der Inhalt von IY und von SP 98DAH, unabhängig vom bisherigen Inhalt von SP.

# PUSH qq

Retten eines Registerpaares in das Stack

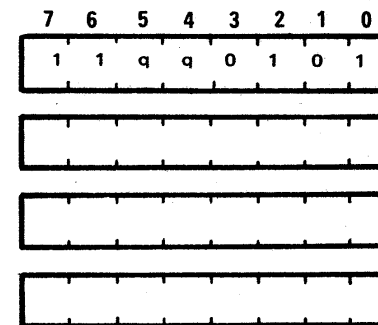
Was macht der Computer?



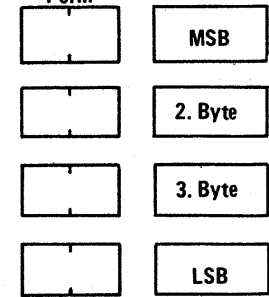
Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	PUSH	qq	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu$ s / 4.4 $\mu$ s	3	11 (5,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt des Registerpaars qq wird in den externen RAM-Stack kopiert, der nach dem LIFO-Prinzip (Last In First Out) organisiert ist. Das Stack-Pointer-Doppelregister SP enthält dabei dauernd die aktuelle 16bit-Adresse (also die niederwertigste als Stack benützte Adresse) dieses Stack-Bereichs.

Der PUSH-Befehl subtrahiert zunächst 1 vom Inhalt des Stackpointers und lädt das höherwertige Byte des Registerpaars qq in die Speicherstelle, die nun durch den Inhalt von SP adressiert ist; danach wird vom Inhalt von SP nochmals 1 abgezogen und das niederwertigere Byte des Registerpaars qq in die Speicherstelle, die jetzt durch den Inhalt von SP adressiert ist, transferiert. Für qq sind die Registerpaarnamen BC, DE, HL und AF zulässig (Behandlung der Indexregister s. folgende Seiten).

Die Registerpaarnamen sind im Maschinencode durch folgende Bitkombinationen codiert.

Registerpaar	BC	DE	HL	AF
qq	00	01	10	11

Beispiel: Das Registerpaar AF enthalte den Wert 2233H und SP den Wert 1007H.

Nach Ausführung des Befehls

PUSH AF

enthält Speicherstelle 1006H den Wert 22H, Speicherstelle 1005H den Wert 33H und SP den Wert 1005H.

# PUSH IX

Retten eines Indexregisters in das Stack

Was macht der Computer?

$(SP-2) \leftarrow XI_L, (SP-1) \leftarrow XI_H$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	PUSH	IX	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	0	1	1	1	0	1

1	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

D D	MSB
-----	-----

E 5	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	3	15 (4,5,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Der Inhalt des Registerpaars IX wird in den externen RAM-Stack kopiert, der nach dem LIFO-Prinzip (Last In First Out) organisiert ist. Das Stack-Pointer-Doppelregister SP enthält dabei dauernd die aktuelle 16bit-Adresse dieses Stack-Bereichs (also die niederwertigste als Stack benützte Adresse).

Der PUSH-Befehl subtrahiert zunächst 1 vom Inhalt des Stackpointers und lädt das höherwertige Byte des Registerpaars IX in die Speicherstelle, die nun durch den Inhalt von SP adressiert ist; danach wird vom Inhalt von SP nochmals 1 abgezogen und das niederwertigere Byte des Registerpaars IX in die Speicherstelle, die jetzt durch den Inhalt von SP adressiert ist, transferiert.

Beispiel: Das Registerpaar IX enthalte den Wert 2233H und SP den Wert 1007H.

Nach Ausführung des Befehls

PUSH IX

enthält Speicherstelle 1006H den Wert 22H, Speicherstelle 1005H den Wert 33H und SP den Wert 1005H.

# PUSH IX

Retten eines Indexregisters in das Stack

(SP-2) ← IY<sub>L</sub>, (SP-1) ← IY<sub>H</sub>

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	PUSH	IY	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1

1	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

F D	MSB
-----	-----

E 5	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	3	15 (4,5,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt des Registerpaars IY wird in den externen RAM-Stack kopiert, der nach dem LIFO-Prinzip (Last In First Out) organisiert ist. Das Stack-Pointer-Doppelregister SP enthält dabei dauernd die aktuelle 16bit-Adresse dieses Stack-Bereichs (also die niederwertigste als Stack benützte Adresse).

Der PUSH-Befehl subtrahiert zunächst 1 vom Inhalt des Stackpointers und lädt das höherwertige Byte des Registerpaars IY in die Speicherstelle, die nun durch den Inhalt von SP adressiert ist; danach wird vom Inhalt von SP nochmals 1 abgezogen und das niederwertigere Byte des Registerpaars IY in die Speicherstelle, die jetzt durch den Inhalt von SP adressiert ist, transferiert.

Beispiel: Das Registerpaar IY enthalte den Wert 2233H und SP den Wert 1007H.

Nach Ausführung des Befehls

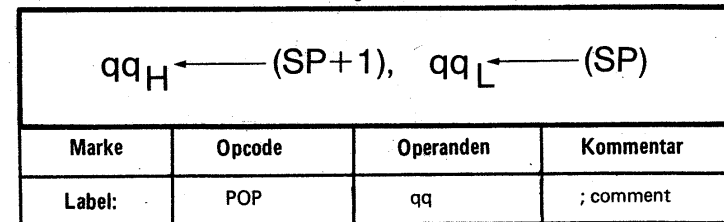
PUSH IY

enthält Speicherstelle 1006H den Wert 22H, Speicherstelle 1005H den Wert 33H und SP den Wert 1005H.

# POP qq

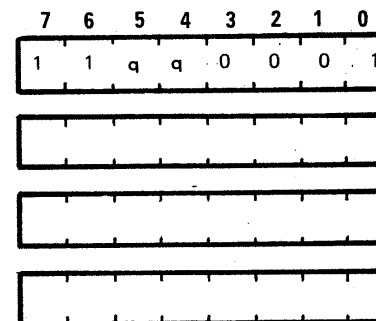
Wiedergewinnen eines Registerpaars aus dem Stack

Was macht der Computer?

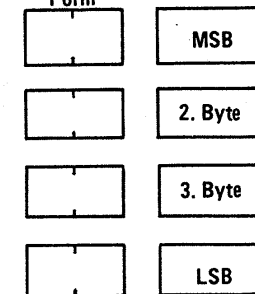


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der vom Stackpointer SP und von (SP-1) adressierten zwei Bytes des externen RAM-Stacks wird in das Registerpaar qq kopiert.

Der POP-Befehl kopiert zunächst den Inhalt der Speicherstelle, die durch den momentanen Inhalt des Stackpointers adressiert ist, in den niederwertigen Teil des Registerpaars qq. Danach wird zum Inhalt von SP 1 addiert und der Inhalt der von seinem neuen Inhalt adressierten Speicherstelle in den höherwertigen Teil des Registerpaars qq kopiert. Daraufhin wird zum Inhalt des Stackpointerregisters SP erneut eine 1 addiert.

Für qq sind in diesem Befehl die Registerpaarnamen BC, DE, HL und AF zulässig, die im Maschinenprogramm durch folgende Bitkombination codiert sind:

Registerpaar	BC	DE	HL	AF
qq	00	01	10	11

Beispiel: Der Stackpointer enthalte den Wert 100H, die durch 100H adressierte Speicherstelle den Wert 55H und Speicherstelle 1001H den Wert 33H.  
Nach Ausführung des Befehls

POP HL

steht dann im Registerpaar der Wert 3355H (unabhängig von dessen bisherigem Inhalt) und im Stackpointer SP der Wert 1002H.

# POP IX

Wiedergewinnen eines Registerpaares aus dem Stack

Was macht der Computer?

$IX_H \leftarrow (SP+1), IX_L \leftarrow (SP)$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	POP	IX	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	0	1	1	1	0	1
1	1	1	0	0	0	0	1

in Sedezimaler Form

D D	MSB
E 1	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu\text{s}$ / 5.6 $\mu\text{s}$	4	14 (4,4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der vom Stackpointer SP und von (SP+1) adressierten zwei Bytes des externen RAM-Stacks wird in das Registerpaar IX kopiert.

Der POP-Befehl kopiert zunächst den Inhalt der Speicherstelle, die durch den momentanen Inhalt des Stackpointers adressiert ist in den niederwertigen Teil des Registerpaars IX. Danach wird zum Inhalt von SP 1 addiert und der Inhalt der von seinem neuen Inhalt adressierten Speicherstelle in den höherwertigen Teils des Registerpaars IX kopiert. Daraufhin wird zum Inhalt des Stackpointerregisters SP erneut eine 1 addiert.

Beispiel: Der Stackpointer enthalte den Wert 1000H, die durch 1000H adressierte Speicherstelle den Wert 55H und Speicherstelle 1001H den Wert 33H.  
Nach Ausführung des Befehls

POP HL

steht dann im Registerpaar der Wert 3355H (unabhängig von dessen bisherigem Inhalt) und im Stackpointer SP der Wert 1002H.

# POP IX

Wiedergewinnen eines Indexregisters aus dem Stack

Was macht der Computer?

$IY_H \leftarrow (SP+1), IY_L \leftarrow (SP)$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	POP	IY	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	1

in Sedezimaler Form

FD	MSB
E1	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu\text{s}$ / 5.6 $\mu\text{s}$	4	14 (4,4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt der vom Stackpointer SP und von (SP+1) adressierten zwei Bytes des externen RAM-Stacks wird in das Registerpaar IY kopiert.

Der POP-Befehl kopiert zunächst den Inhalt der Speicherstelle, die durch den momentanen Inhalt des Stackpointers adressiert ist in den niederwertigen Teil des Registerpaars IY. Danach wird zum Inhalt von SP 1 addiert und der Inhalt der von seinem neuen Inhalt adressierten Speicherstelle in den höherwertigen Teil des Registerpaars IY kopiert. Daraufhin wird zum Inhalt des Stackpointerregisters SP erneut eine 1 addiert.

Beispiel: Der Stackpointer enthalte den Wert 1000H, die durch 1000H adressierte Speicherstelle den Wert 55H und Speicherstelle 1001H den Wert 33H.  
Nach Ausführung des Befehls

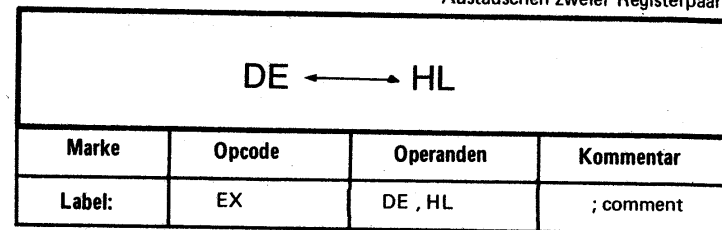
POP HL

steht dann im Registerpaar der Wert 3355H (unabhängig von dessen bisherigem Inhalt) und im Stackpointer SP der Wert 1002H.

# EX DE, HL

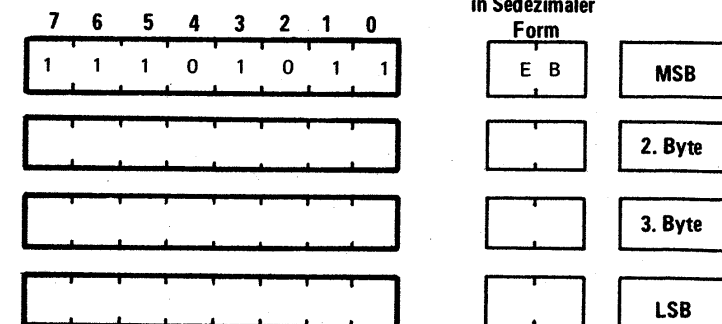
Austauschen zweier Registerpaare

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Die 16bit-Inhalte der Registerpaare DE und HL werden ausgetauscht.

Beispiel: Inhalt von Register D sei 28H, in E stehe der Wert 22H, in H der Wert 49H und in L der Wert 94H.

Nach Ausführung des Befehls

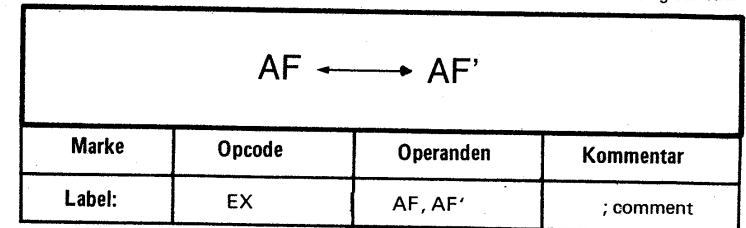
`EX DE,HL`

steht in Register D der Wert 49H, in E der Wert 9AH, in H der Wert 28H und in L der Wert 22H.

# EX AF, AF'

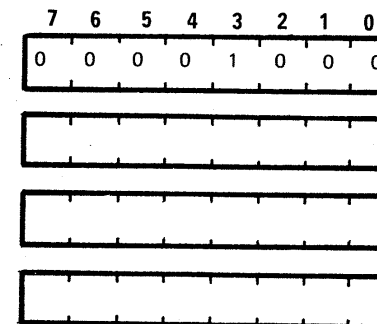
Austausch mit Zweitregistersatz

Was macht der Computer?

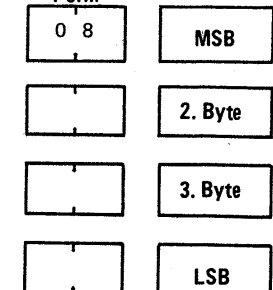


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Die 16bit-Inhalte der Registerpaare AF und AF' werden ausgetauscht (das Registerpaar AF' besteht aus Register A, und Register F')

Beispiel: Register A enthalte den Wert 99H, F den Wert 0H, A' den Wert 59H und F' den Wert 44H.  
Nach Ausführung des Befehls

EX AF,AF'

ist der Inhalt von Register A gleich 59H, F enthält den Wert 44H, A' den Wert 99H und F' den Wert 0H.

# EXX

Austausch mit Zweitregistersatz

Was macht der Computer?

(BC) ↔ (BC'), (DE) ↔ (DE'), (HL) ↔ (HL')

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	EXX		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	0	1	1	0	0	1

in Sedezimaler Form

D 9	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

### Kommentar

Die 16bit-Werte der Registerpaare BC, DE und HL werden mit den 16bit-Werten der Registerpaare BC', DE' und HL' ausgetauscht.

Beispiel:

Register	B	C	D	E	H	L
Inhalt vor Ausführung von EXX	01H	02H	03H	04H	05H	06H
Inhalt nach Ausführung von EXX	07H	08H	09H	10H	11H	12H

Register	B'	C'	D'	E'	H'	L'
Inhalt vor Ausführung von EXX	07H	08H	09H	10H	11H	12H
Inhalt nach Ausführung von EXX	01H	02H	03H	04H	05H	06H

# EX [SP], HL

Austausch zweier Speicherstellen mit einem Registerpaar

Was macht der Computer?

H ← (SP+1), L ← (SP)

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	EX	(SP), HL	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	1

in Sedezimaler Form

E 3	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4,75 $\mu\text{s}$ / 7,6 $\mu\text{s}$	5	19 (4,3,4,3,5)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Der niederwertigere Anteil des Registerpaars HL wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP adressiert ist. Der höherwertige Anteil von HL wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP plus 1 adressiert ist.

Beispiel: Der Inhalt von Registerpaar HL sei 7012H und der von SP sei 8856H. In Speicherstelle 8856H stehe der Wert 11H und in 8857H der Wert 22H.  
Nach Ausführung des Befehls

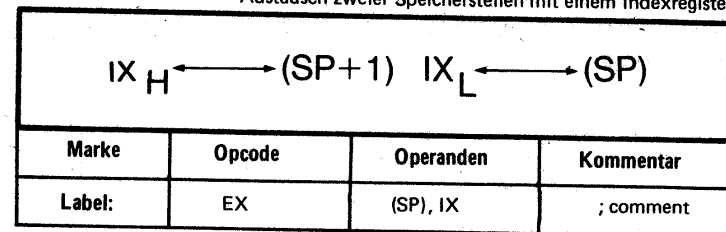
EX (SP), HL

steht im Registerpaar HL der Wert 2211H, Speicherstelle 8856H enthält den Wert 12H, Speicherstelle 8857H den Wert 70H und in Stackpointer steht 8856H.

# EX [SP], IX

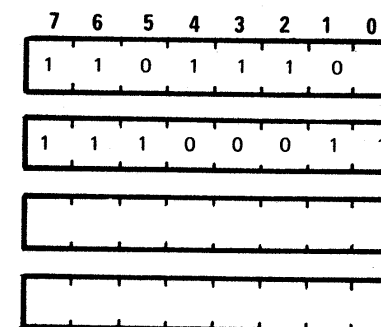
Austausch zweier Speicherstellen mit einem Indexregister

Was macht der Computer?

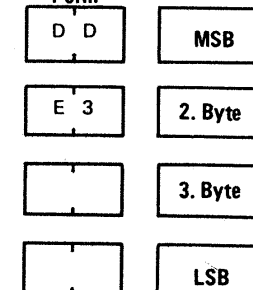


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,4,3,5)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der niederwertigere Anteil des Registerpaars IX wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP adressiert ist. Der höherwertige Anteil von IX wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP plus 1 adressiert ist.

Beispiel: Der Inhalt von Registerpaar IX sei 7012H und der von SP sei 8856H. In Speicherstelle 8856H stehe der Wert 11H und in 8857H der Wert 22H.  
Nach Ausführung des Befehls

EX (SP), IX

steht im Registerpaar IX der Wert 2211H, Speicherstelle 8856H enthält den Wert 12H, Speicherstelle 8857H den Wert 70H und in Stackpointer steht 8856H.

# EX [SP], IY

Austausch zweier Speicherstellen mit einem Indexregister

Was macht der Computer?

$IY_H \longleftrightarrow (SP+1), IY_L \longleftrightarrow (SP)$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	EX	(SP), IY	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

F D	MSB
-----	-----

E 3	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu\text{s}$ / 9.2 $\mu\text{s}$	6	23 (4,4,3,4,3,5)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der niederwertigere Anteil des Registerpaars IY wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP adressiert ist. Der höherwertige Anteil von IY wird gegen den Inhalt der Speicherstelle ausgetauscht, die durch den Inhalt des Stackpointers SP plus 1 adressiert ist.

Beispiel: Der Inhalt von Registerpaar IY sei 7012H und der von SP sei 8856H. In Speicherstelle 8856H stehe der Wert 11H und in 8857H der Wert 22H.  
Nach Ausführung des Befehls

EX (SP), IY

steht im Registerpaar IY der Wert 12H, Speicherstelle 8857H den Wert 70H und im Stackpointer steht 8856H.



# Blocktransfer- und

## Suchbefehle

Quell-

Ziel-

		REG. INDIR.	
		(HL)	
REG. INDIR.	(DE)	ED A0	'LDI' - Load (DE) ← (HL) Inc HL & DE, Dec BC
		ED B0	'LDIR,' - Load (DE) ← (HL) Inc HL & DE, Dec BC, Repeat until BC = 0
		ED A8	'LDD' - Load (DE) ← (HL) Dec HL & DE, Dec BC
		ED B8	'LDDR' - Load (DE) ← (HL) Dec HL & DE, Dec BC, Repeat until BC = 0

Registerpaar HL zeigt auf Quell-Adresse  
Registerpaar DE zeigt auf Ziel-Adresse  
Registerpaar BC wird als Byte-Zähler benutzt

## Block-Transfer-Befehle

REG. INDIR.	(HL)
ED A1	'CPI' Inc HL, Dec BC
ED B1	'CPIR', Inc HL, Dec BC repeat until BC = 0 or find match
ED A9	'CPD' Dec HL & BC
ED B9	'CPDR' Dec HL & BC Repeat until BC = 0 or find match

### Block-Such-Befehle

Registerpaar HL zeigt auf die Speicherstelle, die mit Akkuinhalt verglichen werden soll.

Registerpaar BC wird als Byte-Zähler benutzt.

# LDI

Blocktransportbefehl

Was macht der Computer?

(DE) ← (HL), DE ← DE + 1,  
HL ← HL + 1, BC ← BC - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LDI		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	0	0	0	0	0

in Sedezimaler Form

E D	MSB
A 0	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s/ 6.4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Gesetzt, falls BC-1 ungleich 0, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Befehl kann z.B. zur Übertragung eines bis zu 64k Byte großen Datenblocks von einem Speicherbereich in einen anderen verwendet werden, wobei zwischen die Übertragung eines jeden Bytes andere Aktivitäten eingefügt werden können; außerdem sind beliebige andere Datentransfers möglich.

Hierzu wird der 8bit-Inhalt von der Speicherstelle, die durch den Inhalt des HL-Registerpaares adressiert wird, auf die Speicherstelle übertragen, die durch den Inhalt des DE-Registerpaars adressiert wird. Daraufhin wird zum Inhalt der Registerpaare HL und DE jeweils 1 dazugezählt und vom Inhalt des Registerpaars (Byte-Zähler) BC der Wert 1 abgezogen.

Beispiel:

Folgende Zustände seien vorausgesetzt.

Name	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Reg.Paar
	HL	DE	2222H	1111H	BC
Inhalt	1111H	2222H	66H	88H	0007H

Nach Ausführung des Befehls

LDIR

sind folgende Verhältnisse gegeben:

Name	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Reg.Paar
	HL	DE	2222H	1111H	BC
Inhalt	1112H	2223H	88H	88H	0006H

# LDIR

Blocktransferbefehl

Was macht der Computer?

(DE) ← (HL), DE ← DE + 1,  
HL ← HL + 1, BC ← BC - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	LDIR		; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	1	0	0	0	0

in Sedezimaler Form

E D	MSB
B 0	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Bei BC ≠ 0  
Bei BC = 0

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.25 $\mu$ s / 8.4 $\mu$ s	5	21 (4,4,3,5,5)
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

# LDD

Der Befehl kann z.B. zur Übertragung eines bis zu 64k Byte großen Datenblocks von einem Speicherbereich in einen anderen verwendet werden. Hierzu wird der 8bit-Inhalt von der Speicherstelle, die durch den Inhalt des HL-Registerpaares adressiert wird, auf die Speicherstelle übertragen, die durch den Inhalt des DE-Registerpaares adressiert wird. Daraufhin wird zum Inhalt der Registerpaare HL und DE jeweils 1 dazugezählt und vom Inhalt des Registerpaares (Byte-Zähler) BC der Wert 1 abgezogen.

Falls hierbei im Registerpaar BC der Wert Null erreicht ist, ist die Durchführung des Befehls abgeschlossen. Andernfalls wird vom Inhalt des Befehlszählers der Wert 2 abgezogen, wodurch die gleiche Anweisung erneut zur Ausführung gebracht wird. Durch diese Form der Befehlsausführung, bei der der Befehlscode vom Programm-Speicher dauernd neu eingelesen wird, wird erreicht, daß Interrupts und DMA nicht erst nach Übertragung des gesamten Speicherblocks, sondern nach Übertragung eines jeden Bytes möglich ist.

Zur Sicherstellung des Refresh werden außerdem nach der Übertragung eines jeden Bytes 2 Refreshzyklen eingefügt.

Zur Beachtung: Hat das Registerpaar BC vor Ausführung des Befehls LDIR den Inhalt Null, so erfolgt die Ausführung einer Übertragungsschleife über den gesamten 64k Byte Speicherbereich!

Beispiel zur Befehlsausführung:

Folgende Verhältnisse seien gegeben.

Name	Reg.Paar	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	DE	BC	1111H	1112H	1113H	2222H	2223H	2224H
Inhalt	1111H	2222H	0003H	88H	36H	A5H	66H	59H	C5H

Nach vollständiger Ausführung des Befehls

LDIR

ergibt sich dann

Name	Reg.Paar	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	DE	BC	1111H	1112H	1113H	2222H	2223H	2224H
Inhalt	1114H	2225H	0000H	88H	36H	A5H	88H	36H	A5H

\* Für den Fall, daß Inhalt von BC ≠ 0.

Andernfalls gleiche Befehlsausführungscharakteristika wie Befehl LDI.

Blocktransferbefehl

Was macht der Computer?

(DE) ← (HL), DE ← DE - 1,			
HL ← HL - 1, BC ← BC - 1			
Marke	Opcode	Operanden	Kommentar
Label:	LDD		; comment

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	0	1	0	0	0

in Sedezimaler Form

E D	MSB
A 8	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s/ 6.4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Gesetzt, falls BC-1 ≠ 0, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Befehl kann z.B. zur Übertragung eines bis zu 64k Byte großen Datenblocks von einem Speicherbereich in einen anderen verwendet werden, wobei zwischen die Übertragung eines jeden Bytes andere Aktivitäten eingefügt werden können; außerdem sind beliebige andere Datentransfers möglich. Hierzu wird der 8bit-Inhalt von der Speicherstelle, die durch den Inhalt des HL-Registerpaares adressiert wird, auf die Speicherstelle übertragen, die durch den Inhalt des DE-Registerpaares adressiert wird. Daraufhin wird vom Inhalt der Registerpaare HL und DE jeweils 1 abgezogen und vom Inhalt des Registerpaares (Byte-Zähler) BC der Wert 1 abgezogen.

Beispiel:

Folgende Zustände seien vorausgesetzt:

Name	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Reg.Paar
	HL	DE	2222H	1111H	BC
Inhalt	1111H	2222H	66H	88H	0007H

Nach Ausführung des Befehls

LDD

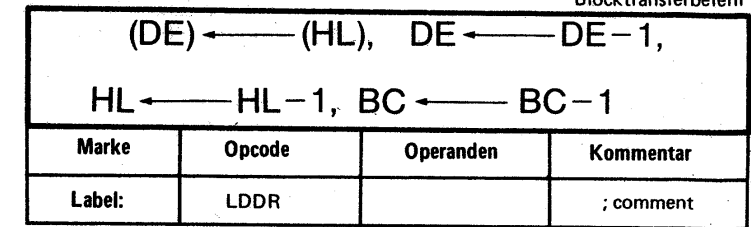
sind folgende Verhältnisse gegeben:

Name	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Reg.Paar
	HL	DE	2222H	1111H	BC
Inhalt	1110H	2221H	88H	88H	0006H

# LDDR

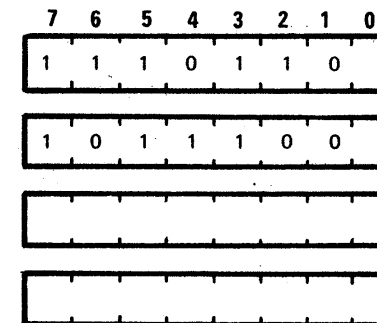
Blocktransferbefehl

Was macht der Computer?

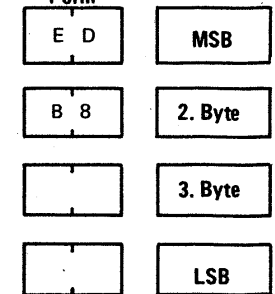


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

	Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
Bei BC $\neq$ 0	5.25 $\mu\text{s}$ / 8.4 $\mu\text{s}$	5	21 (4,4,3,5,5)
Bei BC = 0	4 $\mu\text{s}$ / 6.4 $\mu\text{s}$	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

Der Befehl kann z.B. zur Übertragung eines bis zu 64k Byte großen Datenblocks von einem Speicherbereich in einen anderen verwendet werden. Hierzu wird der 8bit-Inhalt von der Speicherstelle, die durch den Inhalt des HL-Registerpaares adressiert wird, auf die Speicherstelle übertragen, die durch den Inhalt des DE-Registerpaares adressiert wird. Daraufhin wird vom Inhalt der Registerpaare HL und DE jeweils 1 abgezogen und vom Inhalt des Registerpaares (Byte-Zähler) BC der Wert 1 abgezogen.

Falls hierbei im Registerpaar BC der Wert Null erreicht ist, ist die Durchführung des Befehls abgeschlossen. Andernfalls wird vom Inhalt des Befehlszählers der Wert 2 abgezogen, wodurch die gleiche Anweisung erneut zur Ausführung gebracht wird. Durch diese Form der Befehlsausführung, bei der der Befehlsopcode vom Programm-Speicher dauernd neu eingelesen wird, wird erreicht, daß Interrupts und DMA nicht erst nach Übertragung des gesamten Speicherblocks, sondern nach Übertragung eines jeden Bytes möglich ist.

Beispiel zur Befehlsausführung:

Folgende Verhältnisse seien gegeben

Name	Reg.Paar	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	DE	BC	1111H	1110H	110FH	2222H	2221H	2220H
Inhalt	1111H	2222H	0003H	88H	36H	A5H	66H	59H	C5H

Nach vollständiger Ausführung des Befehls

LDDR

ergibt sich dann:

Name	Reg.Paar	Reg.Paar	Reg.Paar	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	DE	BC	1111H	1110H	110FH	2222H	2221H	2220H
Inhalt	110FH	2220H	0000H	88H	36H	A5H	88H	36H	A5H

# CPI

Blocksuchbefehl

Was macht der Computer?

A ← (HL), HL ← HL + 1,

BC ← BC - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	CPI		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1

1	0	1	0	0	0	0	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

E D	MSB
-----	-----

A 1	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei A = (HL), sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei BC-1 $\neq$ 0, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst



## Kommentar

Der Befehl kann z.B. zum Aufsuchen eines bestimmten Zeichens im Arbeitsspeicherbereich von 64k Byte verwendet werden, wobei zwischen der Prüfung eines jeden Bytes andere Aktivitäten eingefügt werden können.

Hierzu wird der Inhalt der Speicherstelle, die durch den Inhalt von HL adressiert ist, mit dem Inhalt des Registers A ("Akkumulator") verglichen. Falls beide Inhalte gleich sind, wird das Null-Bedingungsbit gesetzt. Danach wird zum Inhalt von HL der Wert 1 dazugezählt und vom Inhalt des Registerpaars BC (= Byte-Zähler) der Wert 1 abgezogen.

Beispiel zur Befehlsausführung.

Folgende Zustände seien vorausgesetzt:

Name	Reg.Paar HL	Reg.Paar BC	Reg. A	Speicherstelle 1111H
Inhalt	1111H	0001H	3BH	3BH

Nach vollständiger Ausführung des Befehls

CPI

sind dann folgende Verhältnisse gegeben:

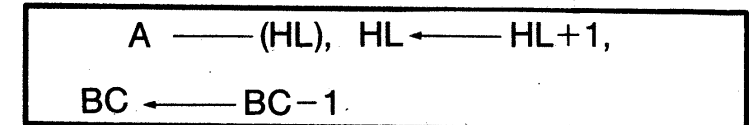
Name	Reg.Paar HL	Reg.Paar BC	Reg. A	Speicherstelle 1111H
Inhalt	1112H	0000H	3BH	3BH

Außerdem ist das Nullbedingungsbit Z (= Zero-Flag") im Register F gesetzt und das P/V-Bedingungsbit im Register F rückgesetzt.

# CPIR

Blocksuchbefehl

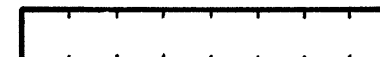
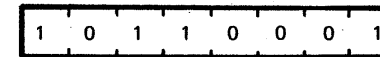
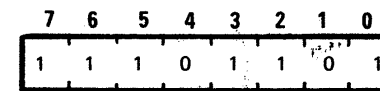
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	CPIR		; comment

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Bei BC ≠ 0 u. A ≠ (HL)  
Bei BC = 0 u. A = (HL)

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5,25 $\mu$ s / 8,4 $\mu$ s	5	21 (4,4,3,5,5)
4 $\mu$ s / 6,4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei A = (HL), sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei BC-1 ≠ 0, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst

Der Befehl kann z.B. zum Aufsuchen eines bestimmten Zeichens im Arbeitsspeicherbereich von 64k Byte verwendet werden.

Hierzu wird der Inhalt der Speicherstelle, die durch den Inhalt von HL adressiert ist, mit dem Inhalt des Registers A ("Akkumulator") verglichen. Falls beide Inhalte gleich sind, wird das Null-Bedingungsbit gesetzt. Danach wird zum Inhalt von HL der Wert 1 dazugezählt und vom Inhalt des Registerpaars BC (= Byte-Zähler) der Wert 1 abgezogen.

Falls hierbei der Inhalt von BC gleich Null wird oder falls sich beim Vergleich von aktuell angesprochener Speicherstelle und A Gleichheit ergibt, ist der Befehl vollständig abgearbeitet. Andernfalls wird vom Befehlszähler der Wert 2 abgezogen, wodurch die gleiche Anweisung erneut zur Ausführung gebracht wird.

Durch diese Form der Befehlsausführung, bei der der Befehlsopcode vom Programm-Speicher dauernd neu eingelesen wird, wird erreicht, daß Interrupts und DMA nicht erst nach Durchsuchen des gesamten Speicherblocks, sondern nach Prüfen eines jeden Bytes 2 Refreshzyklen eingefügt.

Zur Sicherstellung des Refresh werden außerdem nach Prüfung eines jeden Bytes 2 Refreshzyklen eingefügt.

Zur Beachtung: Hat das Registerpaar BC vor Ausführung des Befehls CPDR den Inhalt Null, so erfolgt eine Suchschleife über den ganzen 64k Byte-Speicherbereich!

Beispiel zur Befehlsausführung.

Folgende Verhältnisse seien gegeben:

Name	Reg.Paar	Reg.Paar	Reg.	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	BC	A	1111H	1112H	1113H	1114H
Inhalt	1111H	0007H	F3H	52H	00H	F3H	F5H

Nach vollständiger Ausführung des Befehls

CPDR

ergibt sich dann

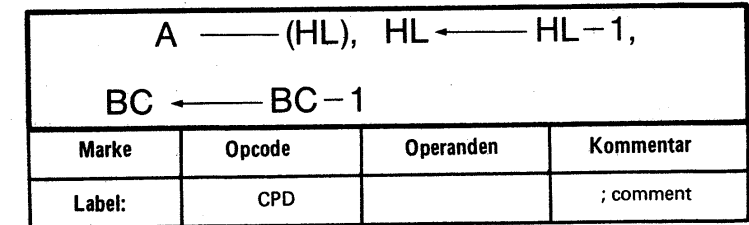
Name	Reg.Paar	Reg.Paar	Reg.	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	BC	A	1111H	1112H	1113H	1114H
Inhalt	1114H	0004H	F3H	52H	00H	F3H	F5H

Außerdem sind das Null-Bedingungsbit ("Zero-Flag") und das P/V-Bedingungsbit im Register F gesetzt.

# CPD

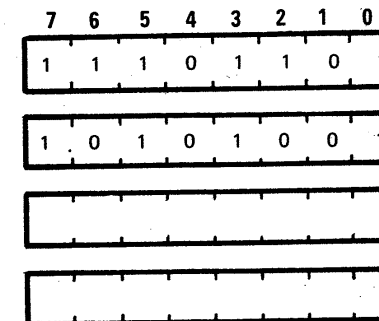
Blocksuchbefehle

Was macht der Computer?

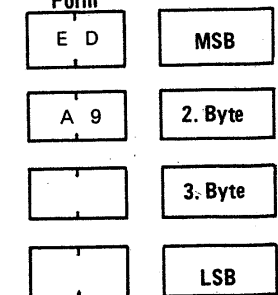


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei A = (HL), sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei BC-1 $\neq$ 0, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Der Befehl kann z.B. zum Aufsuchen eines bestimmten Zeichens im Arbeitsspeicherbereich von 64k Byte verwendet werden, wobei zwischen der Prüfung eines jeden Bytes andere Aktivitäten eingefügt werden können.

Hierzu wird der Inhalt der Speicherstelle, die durch den Inhalt von HL adressiert ist, mit dem Inhalt des Registers A ("Akkumulator") verglichen. Falls beide Inhalte gleich sind, wird das Null-Bedingungsbit gesetzt. Danach wird vom Inhalt von HL der Wert 1 abgezogen und vom Inhalt des Registerpaars BC (= Byte-Zähler) der Wert 1 abgezogen.

Beispiel zur Befehlsausführung.

Folgende Zustände seien vorausgesetzt:

Name	Reg.Paar HL	Reg.Paar BC	Reg. A	Speicherstelle 1111H
Inhalt	1111H	0001H	3BH	3BH

Nach vollständiger Ausführung des Befehls

CPD

sind dann folgende Verhältnisse gegeben:

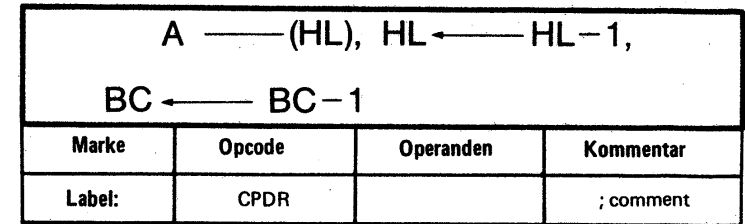
Name	Reg.Paar HL	Reg.Paar BC	Reg. A	Speicherstelle 1111H
Inhalt	1110H	0000H	3BH	3BH

Außerdem ist das Nullbedingungsbit Z (= "Zero-Flag") im Register F gesetzt und das P/V-Bedingungsbit im Register F rückgesetzt.

# CPDR

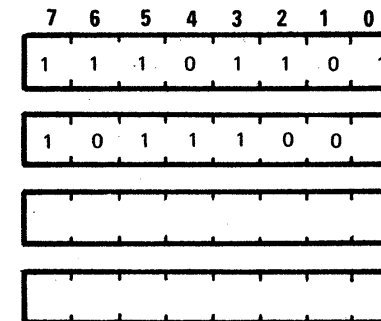
Blocksuchbefehl

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

E D	MSB
B 9	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung \*

	Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
Bei BC $\neq 0$ u. A $\neq$ (HL)	5.25 $\mu\text{s}$ / 8.4 $\mu\text{s}$	5	21 (4,4,3,5,5)
Bei BC = 0 u. A = (HL)	4 $\mu\text{s}$	4	16 (4,4,3,5)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei A = (HL), sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei BC-1 $\neq 0$ , sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst

Der Befehl kann z.B. zum Aufsuchen eines bestimmten Zeichens im Arbeitsspeicherbereich von 64k Byte verwendet werden.

Hierzu wird der Inhalt der Speicherstelle, die durch den Inhalt von HL adressiert ist, mit dem Inhalt des Registers A ("Akkumulator") verglichen. Falls beide Inhalte gleich sind, wird das Null-Bedingungsbit gesetzt. Danach wird vom Inhalt von HL der Wert 1 abgezogen und vom Inhalt des Registerpaares BC (= Byte-Zähler) der Wert 1 abgezogen.

Falls hierbei der Inhalt von BC gleich Null wird oder falls sich beim Vergleich von aktuell angesprochener Speicherstelle und A Gleichheit ergibt, ist der Befehl vollständig abgearbeitet. Andernfalls wird vom Befehlszähler der Wert 2 abgezogen, wodurch die gleiche Anweisung erneut zur Ausführung gebracht wird.

Durch diese Form der Befehlsausführung, bei der der Befehlsopcode vom Programm-Speicher dauernd neu eingelesen wird, wird erreicht, daß Interrupts und DMA nicht erst nach Durchsuchen des gesamten Speicherblocks, sondern nach Prüfen eines jeden Bytes möglich ist. Zur Sicherstellung des Refresh werden außerdem nach Prüfung eines jeden Bytes 2 Refreshzyklen eingefügt.

Zur Beachtung: Hat das Registerpaar BC vor Ausführung des Befehls CPDR den Inhalt Null, so erfolgt eine Suchschleife über den ganzen 64k Byte-Speicherbereich!

Beispiel zur Befehlsausführung.

Folgende Verhältnisse seien gegeben:

Name	Reg.Paar	Reg.Paar	Reg.	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	BC	A	1111H	1110H	110FH	110EH
Inhalt	1111H	0007H	F3H	52H	00H	F3H	F5H

Nach vollständiger Ausführung des Befehls

CPDR

ergibt sich dann

Name	Reg.Paar	Reg.Paar	Reg.	Speicher- stelle	Speicher- stelle	Speicher- stelle	Speicher- stelle
	HL	BC	A	1111H	1110H	110FH	110EH
Inhalt	110EH	0004H	F3H	52H	00H	F3H	F5H

Außerdem sind das Null-Bedingungsbit (= "Zero-Flag") und das P/V-Bedingungsbit im Register F gesetzt.

\* Für den Fall, daß Inhalt von BC ≠ 0 und A ≠ (HL).

Andernfalls gleiche Befehlsausführungscharakteristika wie Befehl CPD.

# 8 bit-Arithmetisch/

## Logische Befehle

Quellregister

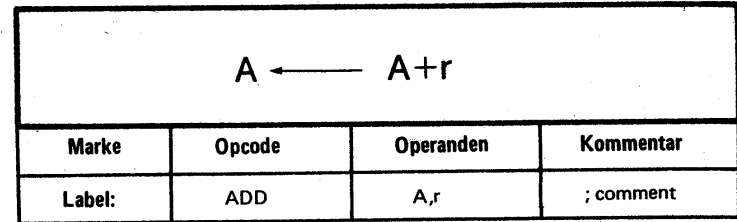
	REGISTER ADDRESSING							REG. INDIR.	INDEXED		IMMED.
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
'ADD'	87	88	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADD w CARRY 'ADC'	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUBTRACT 'SUB'	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SUB w CARRY 'SBC'	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
'AND'	A7	A8	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
'XOR'	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
'OR'	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
COMPARE 'CP'	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INCREMENT 'INC'	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DECREMENT 'DEC'	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

Decimal Adjust Acc, 'DAA'	27
Complement Acc, 'CPL'	2F
Negate Acc, 'NEG' (2's complement)	ED 44

# ADD A, r

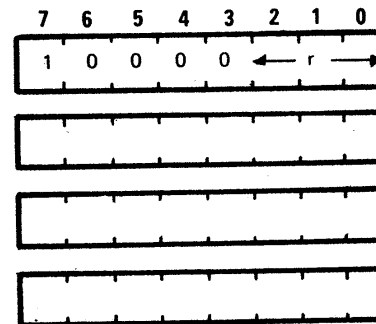
Addition eines Registerinhalts zum Akkuminhalt

Was macht der Computer?

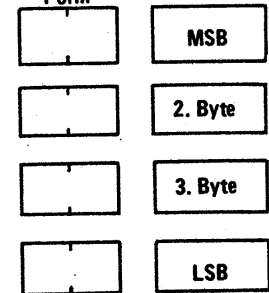


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Der Inhalt des Registers r wird zum Inhalt des Registers A (= Akkumulator) addiert und das Ergebnis im Register A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben. Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H.  
Nach der Ausführung des Befehls

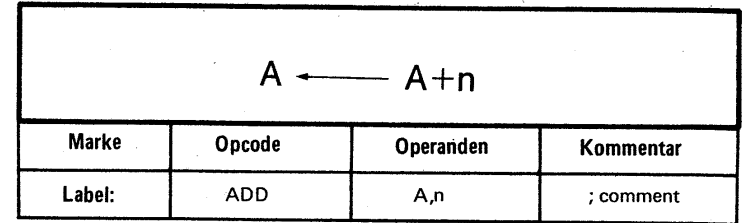
ADD A,C

steht in Register C der Wert 11H und im Akkumulator A der Wert 55H.

# ADD A, n

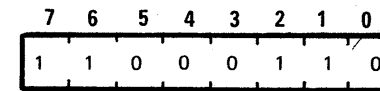
Addition einer Konstanten zum Akkuinhalt

Was macht der Computer?

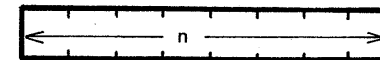


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



C 6	MSB
-----	-----

	2. Byte
--	---------



	3. Byte
--	---------



	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Rückgesetzt
C	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Die 8bit-Konstante n wird zum Inhalt des Registers A (= Akkumulator) addiert und das Ergebnis im Akkumulator abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H.  
Nach Ausführung des Befehls

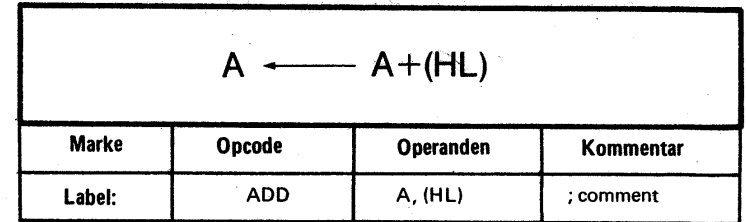
ADD A,33H

hat A den Inhalt 56H.

# ADD A, [HL]

Addition eines Speicherbytes zum Akkumulator

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	0	0	0	0	1	1	0

in Sedezimaler Form

8 6	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird zum Inhalt des Registers A (= Akkumulator) dazugaddiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 01H;

Nach der Ausführung des Befehls

ADD A,(HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert A1H.

# ADD A, [IX+d]

Addition einer indiziert adressierten Speicherstelle zum Akkumulator

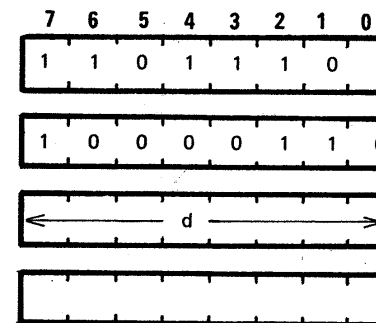
Was macht der Computer?

$$A \leftarrow A + (IX + d)$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	ADD	A, (IX+d)	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

D D	MSB
8 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Rückgesetzt
C	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt



## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz \*d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, zum Inhalt des Registers A (= Akkumulator) addiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

ADD A,(IX+5H)

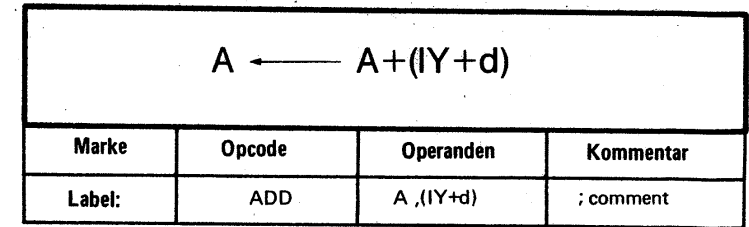
die Inhalte von IX und der Speicherstelle 1005H unverändert;  
in A steht der Wert 33H.

\* Bemerkung: Die Darstellung von d erfolgt in Form des Zweierkomplements.

# ADD A, [IX+d]

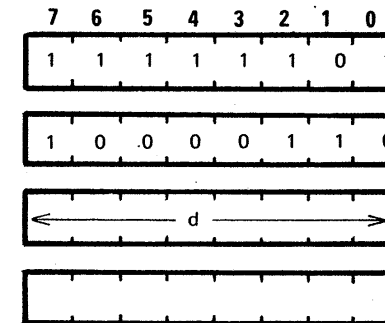
Addition einer indiziert adressierten Speicherstelle zum Akkumulator

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz \*d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, zum Inhalt des Registers A (= Akkumulator) addiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

ADD A,(IY+5H)

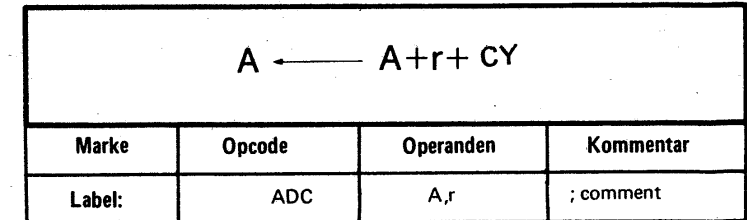
die Inhalte von IY und der Speicherstelle 1005H unverändert;  
in A steht der Wert 33H.

\* Bemerkung: Die Darstellung von d erfolgt in Form des Zweierkomplements.

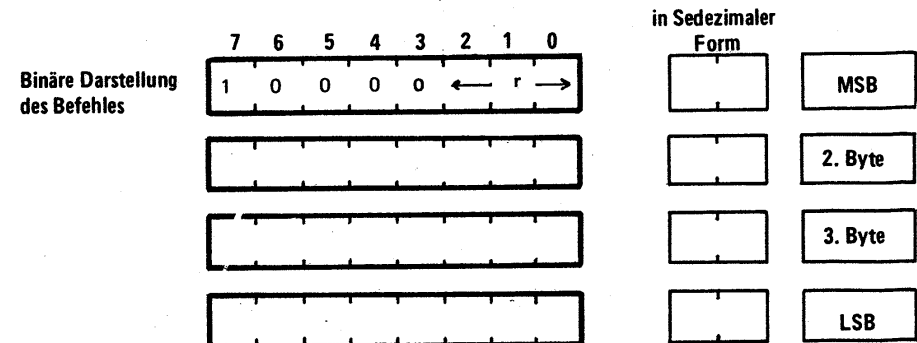
# ADC A, r

Addition eines Registerinhalts plus Carrybit zum Akkuinhalt

Was macht der Computer?



Befehlsformat in Assembler



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Der Inhalt des Registers r wird zum Inhalt des Register A (= Akkumulator) plus dem aktuellen Wert des Übertragsbit (= "Cy") addiert und das Ergebnis im Register A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben. Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H, Register C den Wert 11H und das Überlauf (= Carry)-Bit des Registers F sei gesetzt.  
Nach Durchführung des Befehls

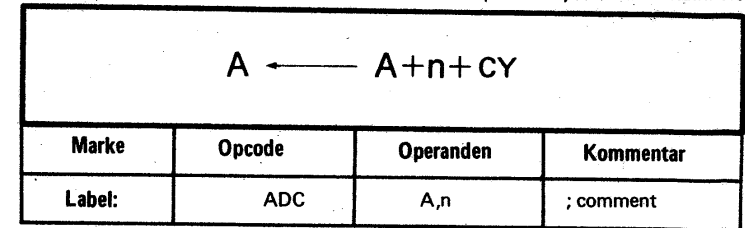
ADC A,C

steht im Register C der Wert 11H, das Überlauf-Bit des Registers F ist rückgesetzt und im Register A steht der Wert 56H.

# ADC A, n

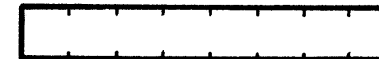
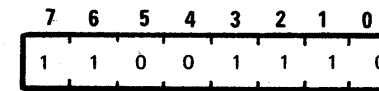
Addition einer Konstanten plus Carrybit zum Akkuinhalt

Was macht der Computer?

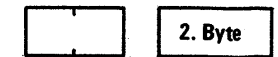


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

### Kommentar

Die 8bit-Konstante n wird zum Inhalt des Registers A (= Akkumulator) plus dem aktuellen Wert des Übertragsbit (= "cy") addiert und das Ergebnis im Akkumulator abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H und das Überlaufbit des Registers F (=cy) sei gesetzt.  
Nach Ausführung des Befehls

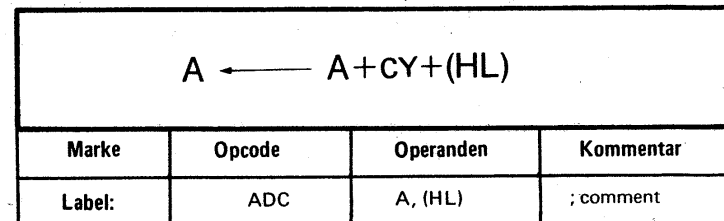
ADC A,33H

hat A den Inhalt 57H und das Überlaufbit ist rückgesetzt.

# ADC A, [HL]

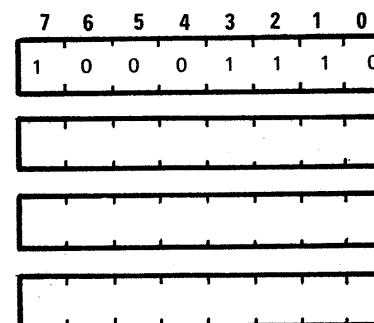
Addition einer Speicherstelle plus Carrybit zum Akkumulator

Was macht der Computer?

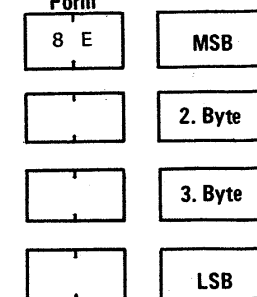


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird zum Inhalt des Registers A (= Akkumulator) plus dem aktuellen Wert des Übertragsbit (cy) dazuaddiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 08H, und das Übertragsbit (=cy) sei gesetzt. Nach Ausführung des Befehls

ADC A,(HL)

steht in der Speicherstelle 2323H der Wert 08H, in A der Wert A9H, und das Übertragsbit ist rückgesetzt.

# ADC A, [IX+d]

Addition einer indiziert adressierten Speicherstelle plus Carrybit zum Akkumulator

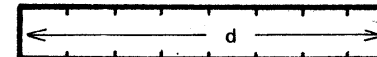
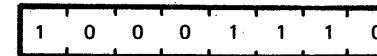
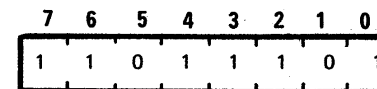
Was macht der Computer?

$$A \leftarrow A + CY + (IX + d)$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	ADC	A, (IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

D D	MSB
-----	-----

8 E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu\text{s}$ / 7.6 $\mu\text{s}$	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Rückgesetzt
C	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, zum Inhalt des Registers A (= Akkumulator) plus dem aktuellen Wert des Übertragsbit (=Cy) dazuaddiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 11H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 22H. Das Übertragsbit (cy) sei gesetzt. Dann sind nach Ausführung von

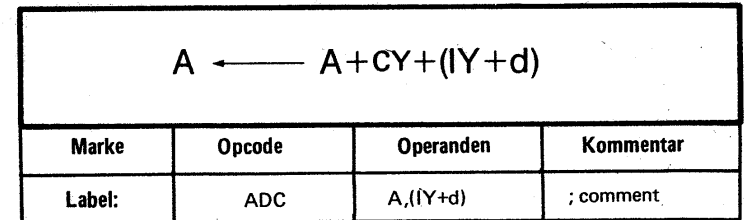
ADC A,(IX+5H)

die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert 34H, das Übertragsbit ist rückgesetzt.

# ADC A, [IX+d]

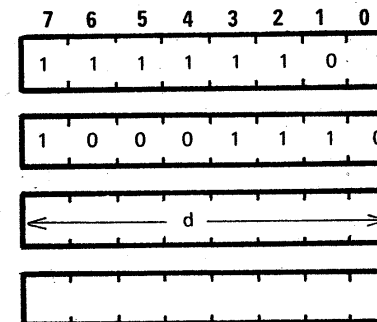
Addition einer indiziert adressierten Speicherstelle plus Carrybit zum Akkumulator

Was macht der Computer?

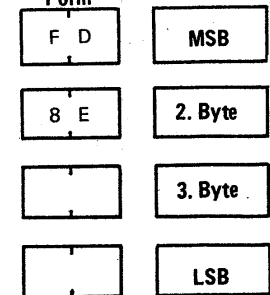


Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, zum Inhalt des Registers A (= Akkumulator) plus dem aktuellen Wert des Übertragsbit (= Cy) dazuaddiert und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 11H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 22H. Das Übertragsbit (cy) sei gesetzt. Dann sind nach Ausführung von

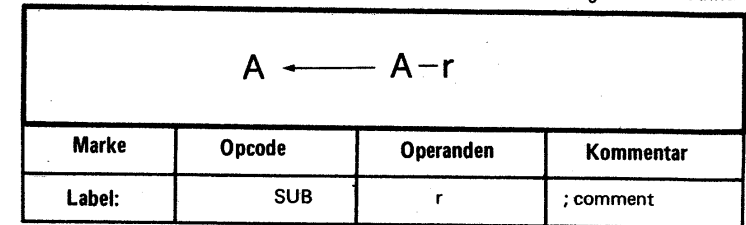
ADC A, (IY+5H)

die Inhalte von IY und der Speicherstelle 1005H unverändert; in A steht der Wert 34H, das Übertragsbit ist rückgesetzt.

# SUB r

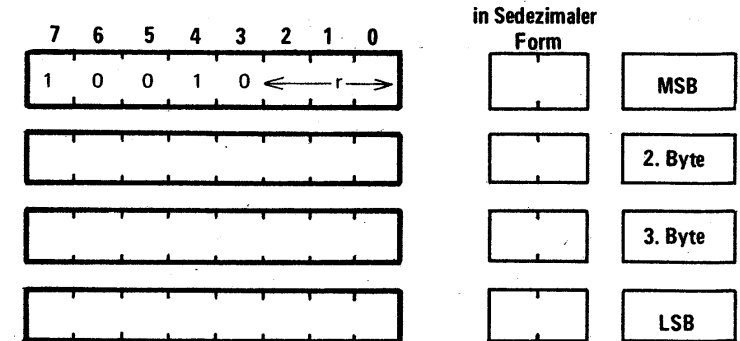
Subtraktion eines Registers vom Akku

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Der Inhalt von Register r wird vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben. Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H.  
Nach Ausführung des Befehls

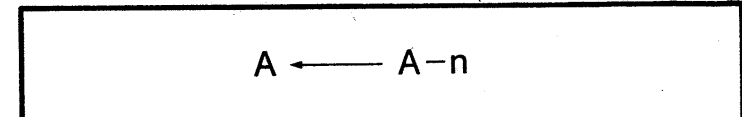
SUB C

steht in Register C der Wert 11H, im Akkumulator A der Wert 33H.

# SUB n

Subtraktion einer Konstanten vom Akku

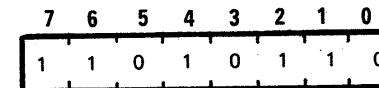
Was macht der Computer?



Befehlsformat in Assembler

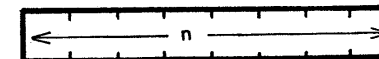
Marke	Opcode	Operanden	Kommentar
Label:	SUB	n	; comment

Binäre Darstellung des Befehles

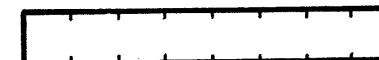


in Sedezimaler Form

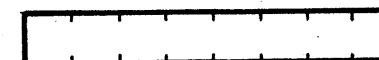
D 6	MSB
-----	-----



	2. Byte
--	---------



	3. Byte
--	---------



	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt



## Kommentar

Die 8bit-Konstante n wird vom Inhalt des Registers A (= Akkumulator) subtrahiert und das Ergebnis im Akkumulator abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H.  
Nach Ausführung des Befehls

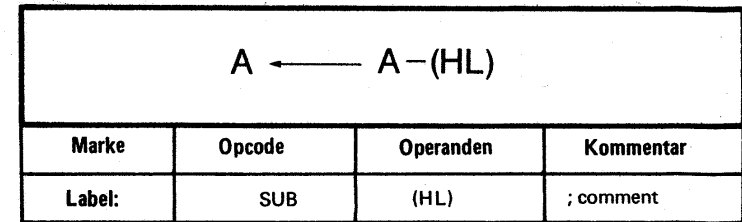
SUB 12H

hat A den Inhalt 11H.

# SUB [HL]

Subtraktion einer Speicherstelle vom Akkumulator

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	0	0	1	0	1	1	0

in Sedezimaler Form

9	6	MSB
		2. Byte
		3. Byte
		LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 01H.

Nach der Ausführung des Befehls

SUB (HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert 9FH.

# SUB [IX+d]

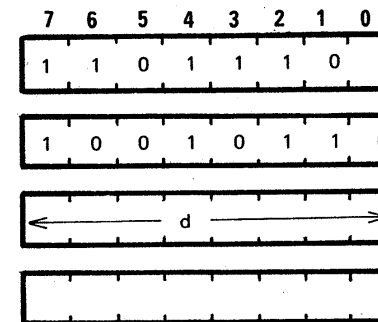
Subtraktion einer indiziert adressierten Speicherstelle vom Akkumulator

$A \leftarrow A - (IX+d)$			
Marke	Opcode	Operanden	Kommentar
Label:	SUB	(IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

D D	MSB
9 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu\text{s}$ / 7.6 $\mu\text{s}$	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz  $d$  addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

SUB (IX+5H)

die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert 11H.

# SUB [IX+d]

Subtraktion einer indiziert adressierten Speicherstelle zum Akkumulator

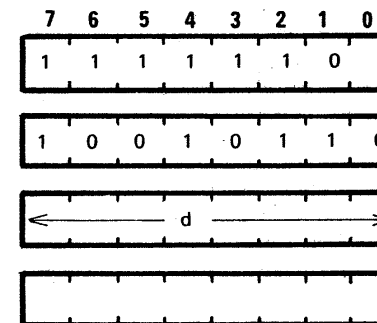
Was macht der Computer?

$$A \leftarrow A - (IX + d)$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SUB	(IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

F D	MSB
9 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19(4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach der Ausführung des Befehls

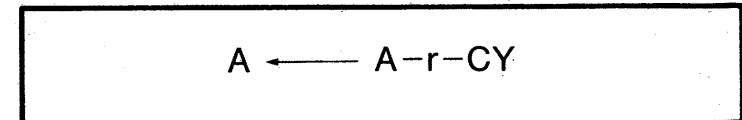
SUB (IY+5H)

die Inhalte von IY und der Speicherstelle 1005H unverändert;  
in A steht der Wert 11H.

# SBC A, r

Subtraktion eines Registers vom Akku mit Carry

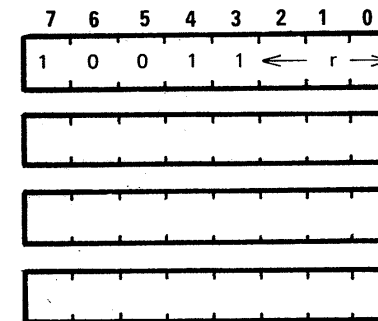
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SBC	A,r	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s/ 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Der Inhalt von Register r und der momentane Wert des Übertragsbedingungsbits (= "Carry") werden vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben. Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H. Das CY-Flag sei gesetzt (= 1).  
Nach der Ausführung des Befehls

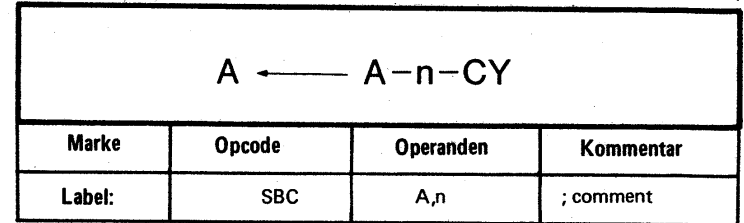
SBC A,C

steht in Register C der Wert 11H, im Akkumulator A der Wert 32H. Das CY-Bedingungsbit ist rückgesetzt.

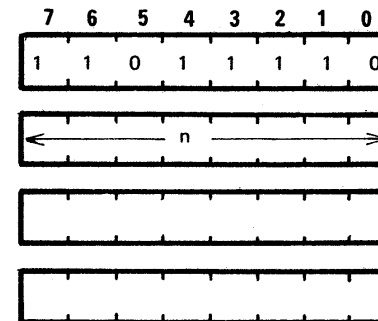
# SBC A, n

Subtraktion einer Konstanten vom Akku mit Carry

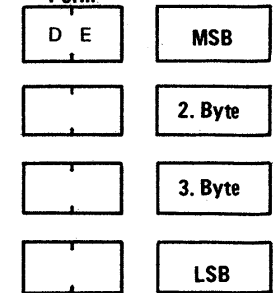
Was macht der Computer?



Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Gesetzt wenn Übertrag von Bit 7, sonst rückgesetzt

## Kommentar

Die 8bit-Konstante n und der aktuelle Wert des Übertragsbit (= "CY") werden vom Inhalt des Registers A (= Akkumulator) subtrahiert und das Ergebnis im Akkumulator abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H und das Überlaufbit des Registers F (= CY) sei gesetzt.

Nach Ausführung des Befehls

SBC A, 11H

hat A den Inhalt 11H und das Übertragsbit ist rückgesetzt.

# SBC A, [HL]

Subtraktion einer Speicherstelle vom Akkumulator mit Carry

Was macht der Computer?

$$A \leftarrow A - (HL) - CY$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SBC	A,(HL)	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	0	0	1	1	1	1	0

in Sedezimaler Form

9 E	MSB
-----	-----

--	--

	2. Byte
--	---------

--	--

	3. Byte
--	---------

--	--

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflußt

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist und der aktuelle Wert des Übertrags-Bedingungsbits (Carry) wird vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H und CY sei gesetzt. In Speicherstelle 2323H stehe der 8bit-Wert 01H; nach der Ausführung des Befehls

SBC A,(HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert 9EH und das CY-Bedingungsbit ist rückgesetzt.

# SBC A, [IX+d]

Subtraktion einer indiziert adressierten Speicherstelle vom Akkumulator mit Carry

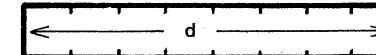
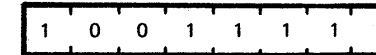
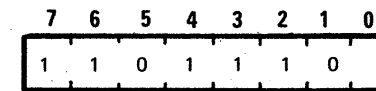
Was macht der Computer?

$$A \leftarrow A - (IX+d) - CY$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SBC	A,(IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

D D	MSB
-----	-----

9 E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflußt

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt bei negativem Übertrag, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz d addiert; dann werden der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird und aktueller Wert des Übertrags-Bedingungsbits (= Carry), vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H, CY sei gesetzt. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

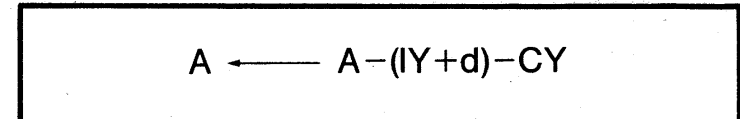
**SBC A,(IX+5H)**

die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert 10H und das CY-Bedingungsbit ist rückgesetzt.

# SBC A, [IY+d]

Subtraktion einer indiziert adressierten Speicherstelle zum Akkumulator mit Carry

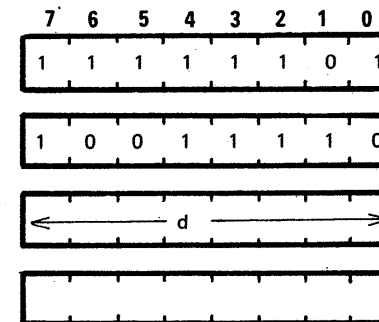
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SBC	A,(IY+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

F D	MSB
9 E	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt bei negativem Übertrag, sonst rückgesetzt



## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann werden der Inhalt der Speicheradresse, die durch die so gebildete Adresse Angesprochen wird und aktueller Wert des Übertragsbedingungsbits (= Carry) vom Inhalt des Registers A (= Akkumulator) abgezogen und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H, CY sei gesetzt. In Speicherstelle 1005H stehe der Wert 11H; dann sind nach Ausführung des Befehls

SBC A,(IY+5H)

die Inhalte von IY und der Speicherstelle 1005H unverändert; in A steht der Wert 10H und das CY-Bedingungsbit ist rückgesetzt.

# AND r

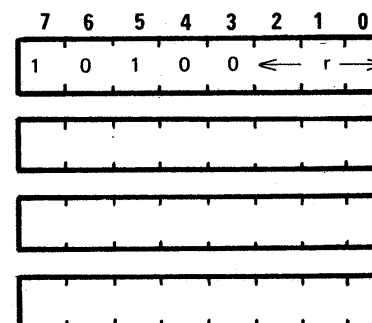
Akku mit Register undieren

Was macht der Computer?

$A \leftarrow A \wedge r$			
Marke	Opcode	Operanden	Kommentar
Label:	AND	r	; comment

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

					MSB				
					2. Byte				
					3. Byte				
					LSB				

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Rückgesetzt

## Kommentar

Der Inhalt des Registers r wird mit dem Inhalt des Registers A (= Akkumulator) undiert und das Ergebnis im Register A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H. Nach Ausführung des Befehls

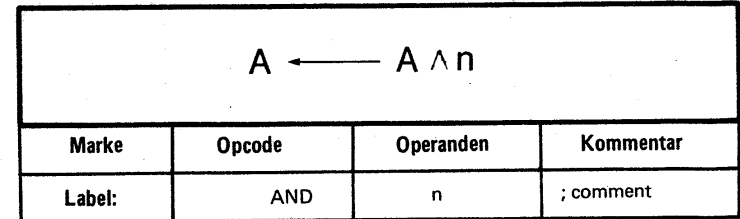
### AND C

steht in Register C der Wert 11H und im Akkumulator A der Wert 00H.

# AND n

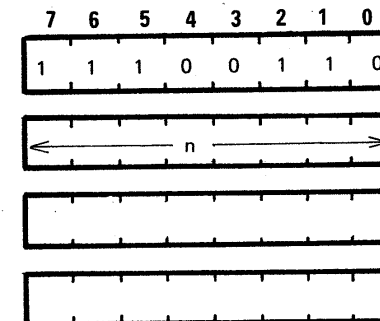
Akku mit Konstante undieren

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

E 6	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Rückgesetzt

## Kommentar

Die 8bit-Konstante n wird mit dem Inhalt des Register A (= Akkumulator) undiert und das Ergebnis im Akkumulator abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H.  
Nach Ausführung des Befehls

AND 33H

hat A den Inhalt 23H.

# AND [HL]

Akku mit Speicherstelle undieren

Was macht der Computer?

$A \leftarrow A \wedge (HL)$			
Marke	Opcode	Operanden	Kommentar
Label:	AND	(HL)	; comment

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	0	1	0	0	1	1	0

in Sedezimaler Form

A 6	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflußt

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird mit dem Inhalt des Registers A (= Akkumulator) undiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 01H;

nach der Ausführung des Befehls

AND (HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert 00H.

# AND [IX+d]

Akku mit indiziert adressierter Speicherstelle undieren

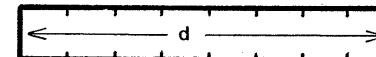
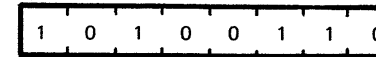
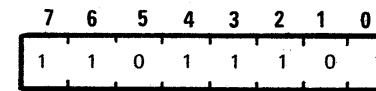
Was macht der Computer?

$$A \leftarrow A \wedge (IX+d)$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	AND	(IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Gesetzt
C	Gesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz  $d$  addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) undiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H.

Nach der Ausführung des Befehls

AND (IX+5)

sind die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert 00H.

# AND [IX+d]

Akku mit indiziert adressierter Speicherstelle undieren

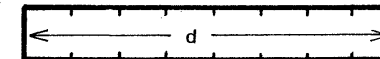
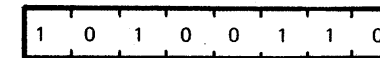
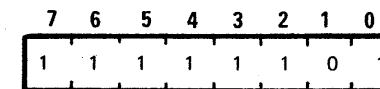
Was macht der Computer?

$$A \leftarrow A \wedge (IX+d)$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	AND	(IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Gesetzt
C	Gesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) undiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H.

Dann sind nach der Ausführung des Befehls

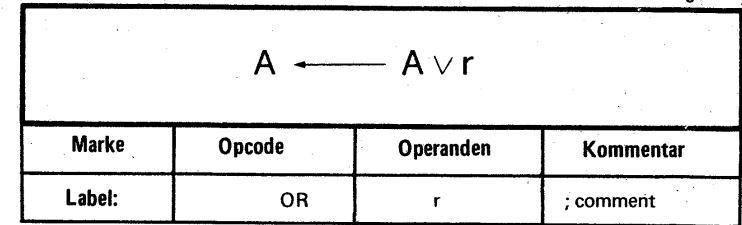
AND (IY+5)

die Inhalte von IY und der Speicherstelle 1005H unverändert; in A steht der Wert 00H.

# OR r

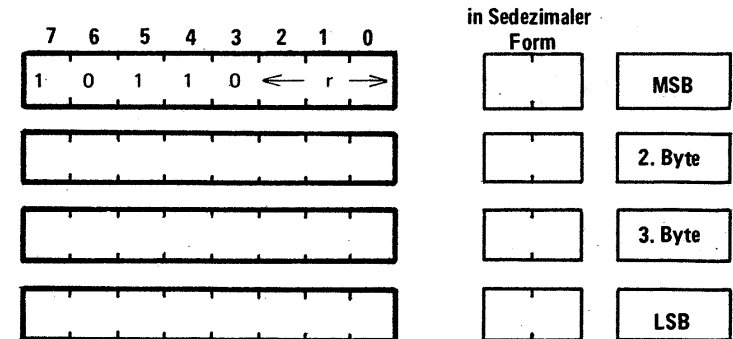
Oderieren des Akkus mit einem Register

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt

## Kommentar

Der Inhalt des Registers r wird mit dem Inhalt des Registers A (= Akkumulator) oderiert und das Ergebnis im Register A abgelegt. Der ursprüngliche Inhalt von A wird dabei überschrieben.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H. Nach Ausführung des Befehls

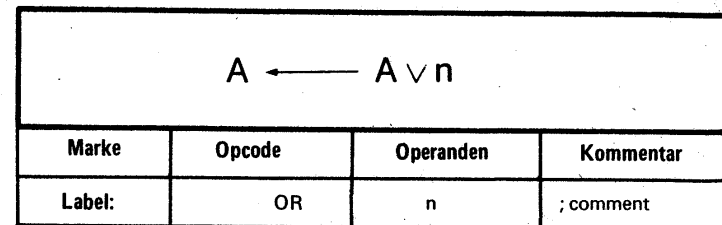
OR C

steht in Register C der Wert 11H und im Akkumulator A der Wert 55H.

# OR n

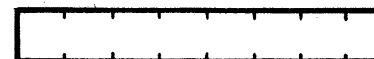
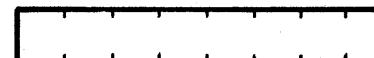
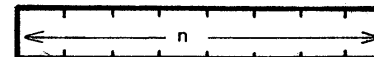
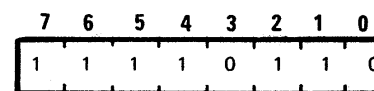
Oderieren des Akkus mit einer Konstanten

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt

## Kommentar

Die 8bit-Konstante n wird mit dem Inhalt des Registers A (= Akkumulator) oderiert und das Ergebnis im Akkumulator abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H.  
Nach Ausführung des Befehls

OR 33H

hat A den Inhalt 33H.

# OR [HL]

Oderieren des Akkus mit einer Speicherstelle

A ← A ∨ (HL)			
Marke	Opcode	Operanden	Kommentar
Label:	OR	(HL)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	0	1	1	0	1	1	0

in Sedezimaler Form

B 6	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Gesetzt
C	Gesetzt



## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird mit dem Inhalt des Registers A (= Akkumulator) oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

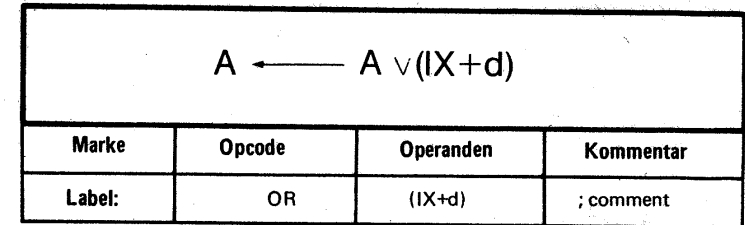
Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bitwert 01H;  
nach der Ausführung des Befehls

OR (HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert A1H.

# OR [IX+d]

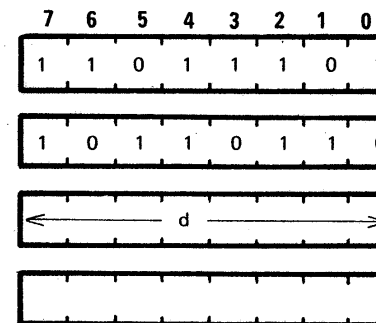
Oderieren des Akkus mit einer indiziert adressierten Speicherstelle



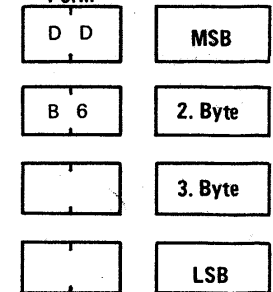
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

OR (IX+5)

die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert 33H.

# OR [IX+d]

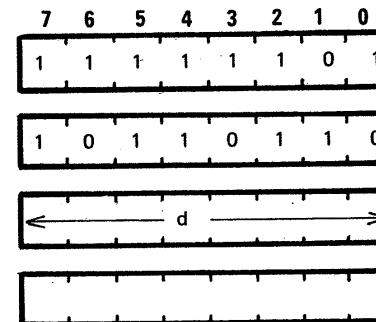
Oderieren des Akkus mit einer indiziert adressierten Speicherstelle

$A \leftarrow A \vee (IX+d)$			
Marke	Opcode	Operanden	Kommentar
Label:	OR	(IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

F D	MSB
B 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4,75 $\mu$ s / 7,6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Gesetzt
C	Gesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H;  
dann sind nach Ausführung des Befehls

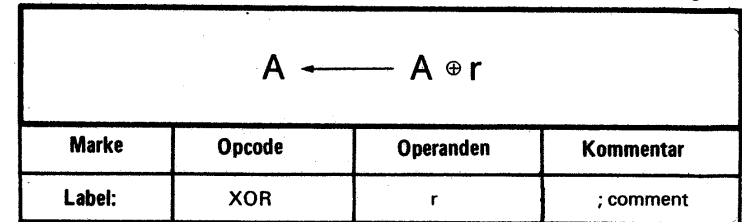
OR (IY+5)

die Inhalte von IY und der Speicherstelle 1005H unverändert; in A steht der Wert 33H.

# XOR r

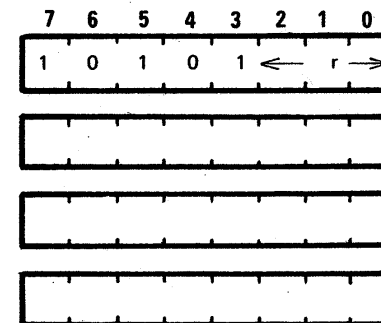
Exklusiv oderieren des Akkus mit einem Register

Was macht der Computer?

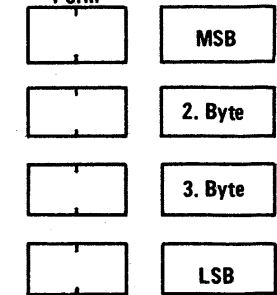


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Rückgesetzt

## Kommentar

Der Inhalt des Registers r wird mit dem Inhalt des Registers A (= Akkumulator) exklusiv oderiert und das Ergebnis im Register A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 44H und Register C den Wert 11H. Nach der Ausführung des Befehls

XOR C

steht in Register C der Wert 11H und im Akkumulator A der Wert 55H.

# XOR n

Exklusiv oderieren des Akkus mit einer Konstanten

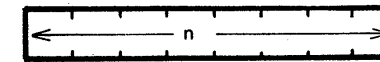
$A \leftarrow A \oplus n$			
Marke	Opcode	Operanden	Kommentar
Label:	XOR	n	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	1	0



in Sedezimaler Form

E E	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Rückgesetzt

## Kommentar

Die 8bit-Konstante n wird mit dem Inhalt des Registers A (= Akkumulator exklusiv oderiert und das Ergebnis im Akkumulator abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert 23H.  
Nach Ausführung des Befehls

XOR 33H

hat A den Inhalt DCH.

# XOR [HL]

Exklusiv oderieren des Akkus mit einer Speicherstelle

$$A \leftarrow A \oplus (HL)$$

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	XOR	(HL)	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	0	1	0	1	1	1	0

in Sedezimaler Form

A E	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird mit dem Inhalt des Registers A (= Akkumulator) exklusiv oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Register A enthalte den Wert A0H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 01H.

Nach der Ausführung des Befehls

XOR (HL)

steht in der Speicherstelle 2323H der Wert 01H und in A der Wert AEH.

# XOR [IX+d]

Exklusiv oderieren des Akkus mit indiziert adressierter Speicherstelle

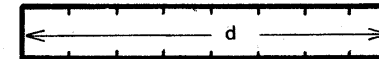
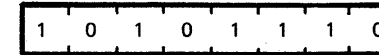
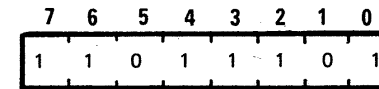
$$A \leftarrow A \oplus (IX+d)$$

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	XOR	(IX+d)	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

D D	MSB
-----	-----

A E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) exklusiv oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H.

Nach der Ausführung des Befehls

XOR (IX+5)

sind dann die Inhalte von IX und der Speicherstelle 1005H unverändert; in A steht der Wert CCH.

# XOR [IX+d]

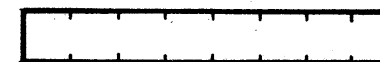
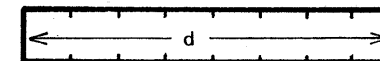
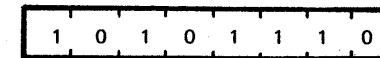
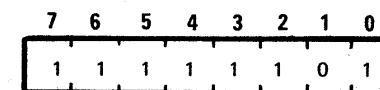
Exklusiv oderieren des Akkus mit einer indiziert adressierten Speicherstelle

$A \leftarrow A \oplus (IX+d)$			
Marke	Opcode	Operanden	Kommentar
Label:	XOR	(IX+d)	; comment

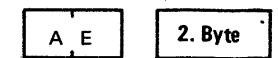
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt
<b>P/V</b>	Gesetzt bei gerader Parität, sonst rückgesetzt
<b>N</b>	Rückgesetzt
<b>C</b>	Rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) exklusiv oderiert und das Ergebnis in A abgelegt.

Der ursprüngliche Inhalt von A wird dabei überschrieben.

Beispiel: Der Inhalt von Register A sei 22H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 11H.

Nach der Ausführung des Befehls

XOR (IY+5)

sind dann die Inhalte von IY und der Speicherstelle 1005H unverändert; in A steht der Wert CCH.

# CP r

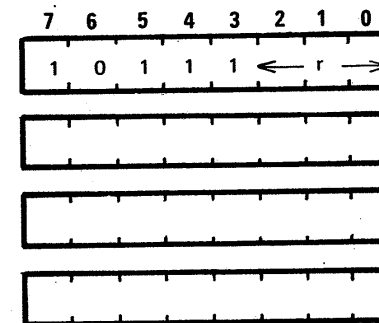
Vergleichen von Akku und Registerinhalt

A-r			
Marke	Opcode	Operanden	Kommentar
Label:	CP	r	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s/ 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt bei Überlauf, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Gesetzt wenn negativer Übertrag, sonst rückgesetzt



## Kommentar

Der Inhalt des Registers r wird mit dem Inhalt des Registers A (= Akkumulator) verglichen.

Der ursprüngliche Inhalt von A bleibt erhalten.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt codiert werden:

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

Beispiel: Register A enthalte 63H und Register C den Wert 60H.  
Nach der Ausführung des Befehls

### CP C

steht in Register C der Wert 60H und im Akkumulator A der Wert 63H; außerdem sind Nullbedingungsbit (Zero-Flag) und P/V-Bedingungsbit im Register F rückgesetzt, unabhängig von ihrem bisherigen Zustand.

# CP n

Vergleich des Akkus mit einer Konstanten

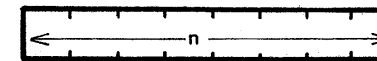
A-n			
Marke	Opcode	Operanden	Kommentar
Label:	CP	n	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	0



in Sedezimaler Form

F E	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt wenn negativer Übertrag, sonst rückgesetzt

## Kommentar

Die 8bit-Konstante n wird mit dem Inhalt des Registers A (= Akkumulator) verglichen.

Der ursprüngliche Inhalt von A bleibt dabei erhalten.

Beispiel: Register A enthalte den Wert 63H.  
Nach der Ausführung des Befehls

CP 60H

hat A den Inhalt 63H, außerdem sind Nullbedingungsbit (Zero-Flag) und P/V-Bedingungsbit im Register F rückgesetzt, unabhängig von ihrem bisherigen Zustand.

# CP [HL]

Vergleich des Akkus mit einer Speicherstelle

A-(HL)			
Marke	Opcode	Operanden	Kommentar
Label:	CP	(HL)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	0

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

BE	MSB
----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt wenn negativer Übertrag, sonst rückgesetzt

## Kommentar

Der 8bit-Inhalt der Speicherstelle, die durch den Inhalt des HL-Registerpaars adressiert ist, wird mit dem Inhalt des Registers A (= Akkumulator) verglichen.

Der ursprüngliche Inhalt von A bleibt dabei erhalten.

Beispiel: Register A enthalte den Wert 63H, das Registerpaar HL den Wert 2323H. In Speicherstelle 2323H stehe der 8bit-Wert 60H.

Nach der Ausführung des Befehls

CP (HL)

steht in der Speicherstelle 2323H der Wert 60H und in A der Wert 63H; außerdem sind Nullbedingungsbit (Zero-Flag) und P/V-Bedingungsbit im Register F rückgesetzt, unabhängig von ihrem bisherigen Zustand.

# CP [IX+d]

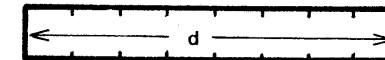
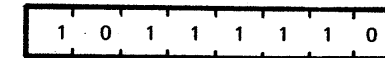
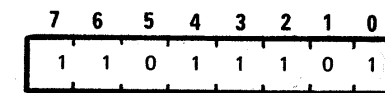
Vergleich einer indiziert adressierten Speicherstelle mit dem Akkumulator

A - (IX+d)			
Marke	Opcode	Operanden	Kommentar
Label:	CP	(IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt wenn negativer Übertrag, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IX und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) verglichen.

Der ursprüngliche Inhalt von A und IX bleibt erhalten.

Beispiel: Der Inhalt von Register A sei 63H und der des Indexregisters IX gleich 1000H. In Speicherstelle 1005H stehe der Wert 60H.

Nach der Ausführung des Befehls

CP (IX+5H)

sind dann die Inhalte von IX, der Speicherstelle 1005H und A unverändert; außerdem sind Nullbedingungsbit (Zero-Flag) und P/V-Bedingungsbit im Register F rückgesetzt, unabhängig von ihrem bisherigen Zustand.

# CP [IX+d]

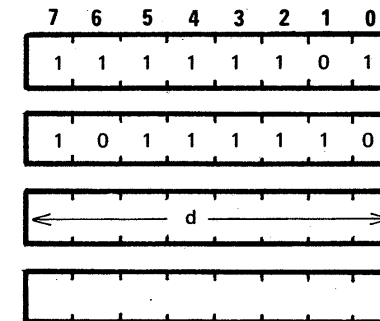
Vergleich einer indiziert adressierten Speicherstelle mit dem Akkumulator

A - (IX+d)			
Marke	Opcode	Operanden	Kommentar
Label:	CP	(IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

F D	MSB
B E	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.75 $\mu$ s / 7.6 $\mu$ s	5	19 (4,4,3,5,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt wenn negativer Übertrag, sonst rückgesetzt

## Kommentar

Bei der Ausführung dieses Befehls werden zunächst der Inhalt des Index-Doppelregisters IY und die Distanz d addiert; dann wird der Inhalt der Speicheradresse, die durch die so gebildete Adresse angesprochen wird, mit dem Inhalt des Registers A (= Akkumulator) verglichen.

Der ursprüngliche Inhalt von A und IY bleibt dabei erhalten.

Beispiel: Der Inhalt von Register A sei 63H und der des Indexregisters IY gleich 1000H. In Speicherstelle 1005H stehe der Wert 60H.

Nach der Ausführung des Befehls

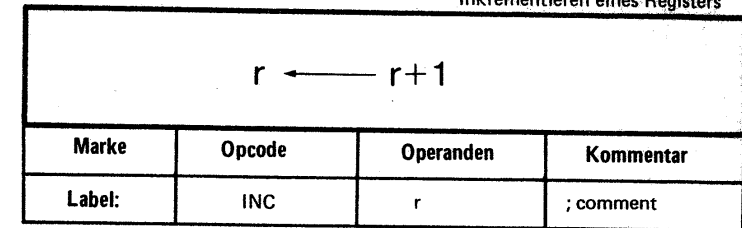
CP (IY+5H)

sind die Inhalte von IY, der Speicherstelle 1005H und A unverändert; außerdem sind Nullbedingungsbit (Zero-Flag) und P/V-Bedingungsbit im Register F rückgesetzt, unabhängig von ihrem bisherigen Zustand.

# INC r

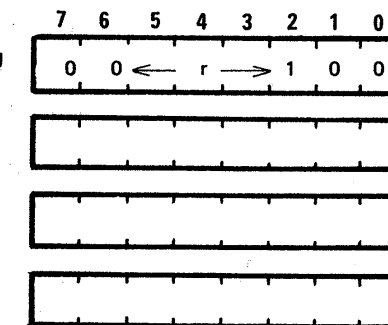
Inkrementieren eines Registers

Was macht der Computer?

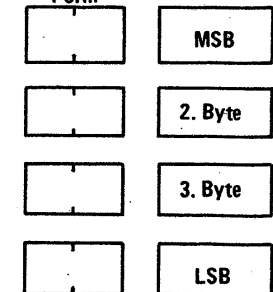


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt, falls r=7FH vor Befehlsausführung war, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Zum Inhalt von Register r wird der Wert 1 dazugezählt.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt verschlüsselt werden.

Registername	A	B	C	D	E	H	L
r	111	000	001	010	011	100	101

Beispiel: Der Inhalt von Register D sei 28H.  
Nach Ausführung des Befehls

INC D

ist der Inhalt des D-Registers 29H.

# INC [HL]

Inkrementieren einer Speicherzelle

$$(HL) \leftarrow (HL) + 1$$

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	INC	(HL)	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	1	1	0	1	0	0

in Sedezimaler Form

3	4	MSB
		2. Byte
		3. Byte
		LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu$ s / 4.4 $\mu$ s	3	11 (4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt, falls r=7FH vor Befehlsausführung, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Zum Inhalt der Speicherstelle, die durch den Inhalt des Registerpaars HL adressiert ist, wird der Wert 1 addiert.

Beispiel: Der Inhalt der Speicherstelle 3434H sei FFH, der Inhalt des Registerpaars HL sei 3434H.  
Nach der Ausführung des Befehls

INC (HL)

enthält die Speicherstelle 3434H den Wert 00H und das Registerpaar HL den Wert 3434H.

# INC [IX+d]

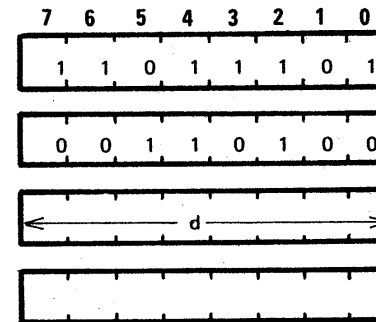
Inkrementieren einer indiziert adressierten Speicherstelle

$(IX+d) \leftarrow (IX+d)+1$			
Marke	Opcode	Operanden	Kommentar
Label:	INC	(IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

D D	MSB
3 4	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt, falls r=7FH vor Befehlsausführung war, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Zum Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IX und der Distanz d adressiert ist, wird der Wert 1 addiert.

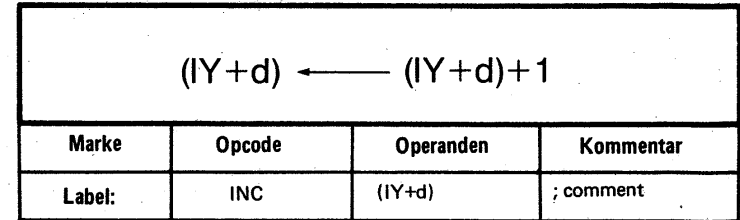
Beispiel: Der Inhalt des Indexdoppelregisters IX sei 2020H. Die Speicherstelle 2030H enthalte den Wert 34H.  
Nach der Ausführung des Befehls

INC (IX+10H)

hat das Indexregister IX den Inhalt 2020H und die Speicherstelle 2030H den Wert 35H.

# INC [IX+d]

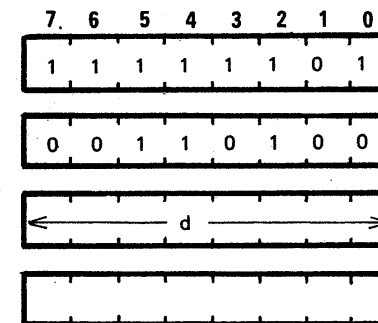
Inkrementieren einer indiziert adressierten Speicherstelle



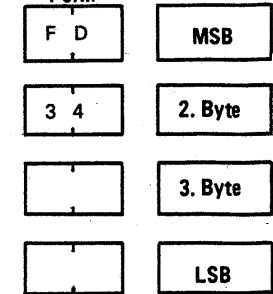
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflußt

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei Übertrag von Bit 3, sonst rückgesetzt
P/V	Gesetzt, falls r=7FH vor Befehlsausführung war, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflußt



## Kommentar

Zum Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IY und der Distanz da adressiert ist, wird der Wert 1 addiert.

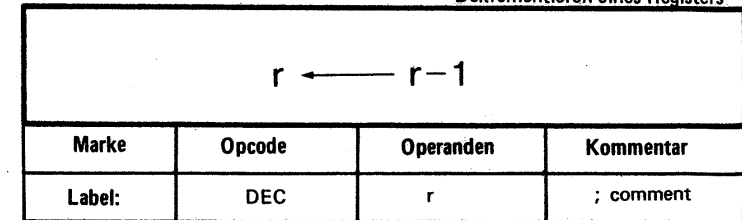
Beispiel: Der Inhalt des Indextoppelregisters IX sei 2020H. Die Speicherstelle 2030H enthalte den Wert 34H. Nach Ausführung des Befehls

INC (IY+10H)

hat das Indexregister IY den Inhalt 2020H und die Speicherstelle 2030H den Wert 35H.

# DEC r

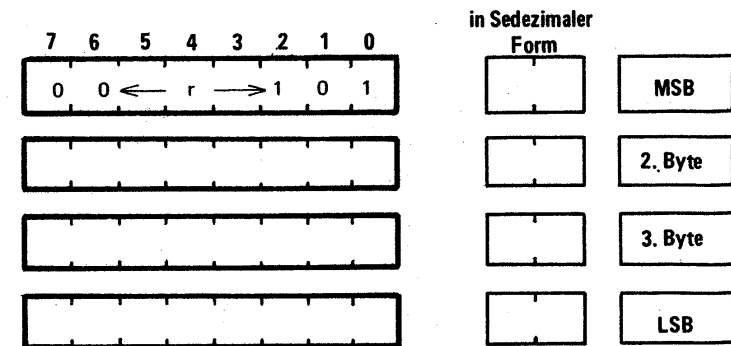
Dekrementieren eines Registers



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

<b>S</b>	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
<b>Z</b>	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
<b>H</b>	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
<b>P/V</b>	Gesetzt, falls Operand zuvor gleich 80H war, sonst rückgesetzt
<b>N</b>	Gesetzt
<b>C</b>	Nicht beeinflusst

## Kommentar

Vom Inhalt von Register r wird der Wert 1 abgezogen.

Für r sind die Registernamen A, B, C, D, E, H und L zulässig, die im binären Maschinencode wie folgt verschlüsselt werden:

Registername	A	B	C	D	E	H	L
r	111	000	001	010	011	100	101

Beispiel: Der Inhalt von Register D sei 28H.  
Nach Ausführung des Befehls

DEC D

ist der Inhalt des D-Registers 27H.

# DEC [HL]

Dekrementieren einer Speicherstelle

(HL) ← (HL) - 1			
Marke	Opcode	Operanden	Kommentar
Label:	DEC	(HL)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	1	1	0	1	0	1

in Sedezimaler Form

3 5	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu$ s / 4.4 $\mu$ s	3	11 (4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt, falls Operand zuvor gleich 80H war, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Vom Inhalt der Speicherstelle, die durch den Inhalt des Registerpaars HL adressiert ist, wird der Wert 1 abgezogen.

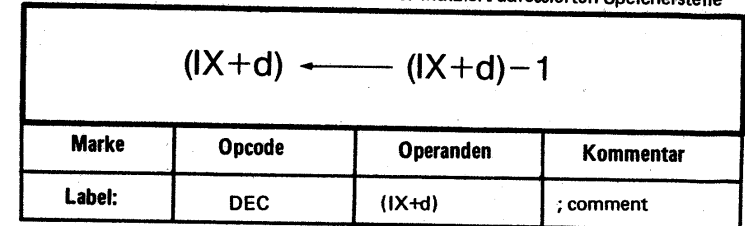
Beispiel: Der Inhalt der Speicherstelle 3434H sei FFH, der Inhalt des Registerpaars HL sei 3434H.  
Nach der Ausführung des Befehls

DEC (HL)

enthält die Speicherstelle 3434H den Wert FEH und das Registerpaar HL den Wert 3434H.

# DEC [IX+d]

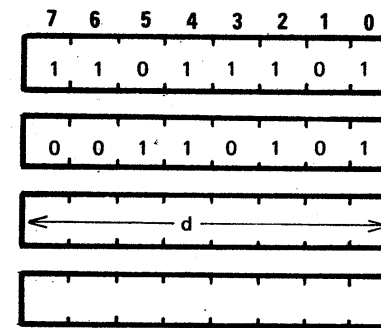
Dekrementieren einer indiziert adressierten Speicherstelle



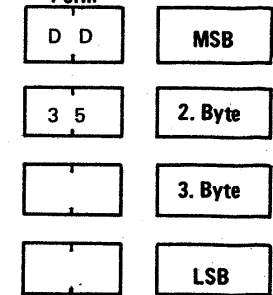
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt, falls Operand zuvor gleich 80H war, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Vom Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IX und der Distanz d adressiert ist, wird der Wert 1 subtrahiert.

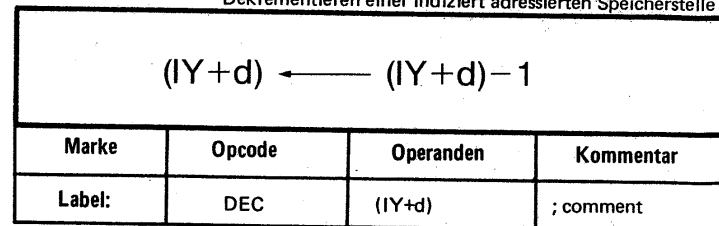
Beispiel: Der Inhalt des Indextoppelregisters IX sei 2020H. Die Speicherstelle 2030H enthalte den Wert 34H. Nach Ausführung des Befehls

DEC (IX+10H)

hat das Indexregister IX den Inhalt 2020H und die Speicherstelle 2030H den Wert 33H.

# DEC [IX+d]

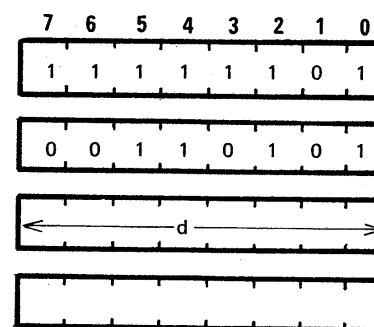
Dekrementieren einer indiziert adressierten Speicherstelle



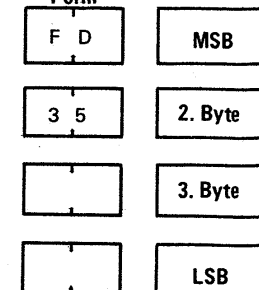
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflußt

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt, falls Operand zuvor gleich 80H war, sonst rückgesetzt
N	Gesetzt
C	Nicht beeinflußt

## Kommentar

Vom Inhalt der Speicherstelle, die durch die Summe aus Indexregisterinhalt IY und der Distanz d adressiert ist, wird der Wert 1 subtrahiert.

Beispiel: Der Inhalt des Indextoppelregisters IY sei 2020H. Die Speicherstelle 2030 enthalte den Wert 34H. Nach Ausführung des Befehls

DEC (IY+10H)

hat das Indexregister IY den Inhalt 2020H und die Speicherstelle 2030H den Wert 33H.

# DAA

Umwandlung des Akkuinhalts in BCD-Format

_____			
Marke	Opcode	Operanden	Kommentar
Label:	DAA		; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
0	0	1	0	0	1	1	1

in Sedezimaler Form

27	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

S	Gesetzt, falls höchstw. Bit von A n. Befehl = 1 ist, sonst rückgesetzt
Z	Gesetzt bei Akku = 0 nach Befehl, sonst rückgesetzt
H	Siehe folgende Tabelle
P/V	Gesetzt bei geradzahligem Ergebnis, sonst rückgesetzt
N	Nicht beeinflusst
C	Siehe folgende Tabelle

### Kommentar

Durch diesen Befehl wird die Umwandlung des Inhalts von A (= Akkumulator) in ein Format durchgeführt, das die Ausführung von BCD-Additionen und Subtraktionen ermöglicht. Zur Durchführung von Additionen über die Befehle ADD, ADC oder INC bzw. Subtraktionen über die Befehle SUB, SBC, DEC oder NEG werden die folgenden Aktivitäten ausgeführt.

Operation	CY vor DAA	HEX-Wert im höherwertigen Halbbyte (Bit 7 - 4)	H vor DAA	HEX-Wert im niederwertigen Halbbyte (Bit 3 - 0)	zum BYTE addierte Zahl	CY nach DAA
ADD ADC INC	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
	0	A-F	0	0-9	60	1
	0	A-F	1	0-3	66	1
SUB SBC DEC NEG	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
	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

Erläuterung: Bei einer arithmetischen Operation zwischen zwei BCD-Zeilen (z.B. 15D und 27D) wird das Ergebnis einfach durch die gewohnte dezimale Addition

$$\begin{array}{r} 15 \\ + 27 \\ \hline 42 \end{array}$$

ermittelt. Bei Zusammenzählen der Binäräquivalente dieser Dezimalzahlen ergibt sich jedoch ein falsches Ergebnis, wie die Ausführung des obigen Beispiels zeigt:

$$\begin{array}{r} 0001 \quad 0101 \quad \hat{=} 15 \text{ H} \\ + 0010 \quad 0111 \quad \hat{=} 27 \text{ H} \\ \hline 0011 \quad 1100 \quad \hat{=} 3C \text{ H} \end{array}$$

Der Befehl DAA nun führt die nötige Korrektur eines solchen Ergebnisses durch, so daß eine richtige Dezimaldarstellung des Ergebnisses einer binären Addition von Dezimalzahlen entsprechend folgendem Beispiel generiert wird:

$$\begin{array}{r} 0011 \quad 1100 \quad = 3C \text{ H} \\ + 0000 \quad 0110 \quad = 27 \text{ H} \\ \hline 0100 \quad 0010 \quad = 42 \text{ H} \end{array}$$

# CPL

Komplementieren des Akkumulators

A ← $\bar{A}$			
Marke	Opcode	Operanden	Kommentar
Label:	CPL		; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	1	0	1	1	1	1

in Sedezimaler Form

2 F	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Gesetzt
P/V	Nicht beeinflusst
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des Registers A (= Akkumulator) wird invertiert (d.h. sein 1er-Komplement gebildet).

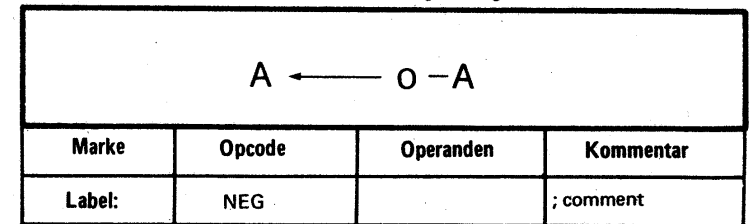
Beispiel: Register A enthalte das Bitmuster '1011 0100.  
Nach Ausführung des Befehls

CPL

ist sein Inhalt 0100 1011.

# NEG

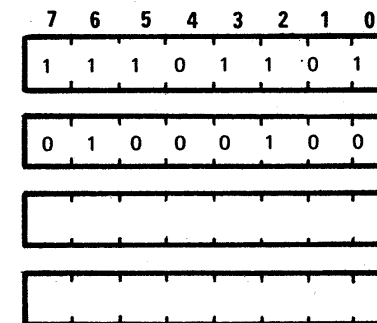
Bildung des negativen Wertes des Akkus



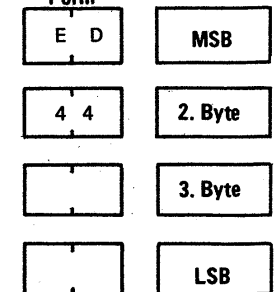
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt bei negativem Ergebnis, sonst rückgesetzt
Z	Gesetzt bei Ergebnis gleich Null, sonst rückgesetzt
H	Gesetzt bei negativem Übertrag von Bit 4, sonst rückgesetzt
P/V	Gesetzt, wenn Inh. des Akkus vor d. Befehl =80H war, s. rückgesetzt
N	Gesetzt
C	Gesetzt, falls Akku $\neq$ 0 vor dem Befehl; sonst rückgesetzt

Kommentar

Es wird der Negativbetrag des Registers A (= Akkumulator) gebildet (also sein Zweierkomplement ermittelt), was der Subtraktion des Akku-Inhalts von Null entspricht.

Bitte beachten Sie, daß bei Akkuinhalt 80H das Ergebnis dieser Operation wieder 80H ist.

Beispiel: Register A enthalte das Bitmuster 1001 1000.  
Nach der Ausführung des Befehls

NEG

ist der Inhalt von A 0110 1000.

# 16 bit-Arithmetisch/

## Logische Befehle

Quellregister

		BC	DE	HL	SP	IX	IY
'ADD'	HL						
	IX	DD 09	DD 19		DD 39	DD 29	
	IY	FD 09	FD 19		FD 39		FD 29
ADD WITH CARRY AND SET FLAGS 'ADC'	HL	ED 4A	ED 5A	ED 6A	ED 7A		
SUB WITH CARRY AND SET FLAGS 'SBC'	HL	ED 42	ED 52	ED 62	ED 72		
INCREMENT 'INC'						DD 23	FD 23
DECREMENT 'DEC'						DD 2B	FD 2B

Zielregister



# ADD HL, SS

Addition eines Registerpaars zu HL

HL ← HL+SS			
Marke	Opcode	Operanden	Kommentar
Label:	ADD	HL,ss	; comment

Was macht der Computer?

Befehlsformat im Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	1	s	s	1	0	1	0

in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu\text{s}$ / 4.4 $\mu\text{s}$	3	11 (4,4,3)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Gesetzt, wenn Übertrag aus Bit 11, sonst rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Gesetzt bei Übertrag aus Bit 15, sonst rückgesetzt

Der Inhalt des Registerpaares ss (BC, DE, HL oder SP) wird zum Inhalt des HL-Registerpaares addiert.

Das Ergebnis steht anschließend in HL.

Registerpaar	BC	DE	HL	SP
ss	00	01	10	11

Beispiel: HL enthalte 4242H und DE enthalte 1111H.  
Nach der Befehlsdurchführung enthält HL 5353H.

# ADC HL, SS

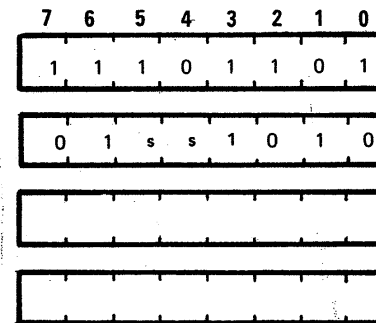
Addition eines Registerpaares mit Carry zu HL

HL ← HL + SS + CY			
Marke	Opcode	Operanden	Kommentar
Label:	ADC	HL,ss	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

E D	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Gesetzt bei Übertrag aus Bit 11, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Rückgesetzt
C	Gesetzt bei Übertrag aus Bit 15, sonst rückgesetzt

## Kommentar

Der Inhalt des Registerpaares ss (eines der Paare BC, DE, HL, SP) wird zum Inhalt von HL addiert, zusätzlich das CY-Bit.

Das Ergebnis der Addition steht anschließend in HL.

Registerpaar	BC	DE	HL	SP
ss	00	01	10	11

Beispiel: BC enthalte 2222H, HL enthalte 5437H, CY sei gesetzt.  
Nach der Befehlsausführung enthält HL 765AH.

# SBC HL, ss

Subtraktion eines Registerpaares von HL mit Carry

HL ← HL - ss - CY			
Marke	Opcode	Operanden	Kommentar
Label:	SBC	HL,ss	; comment

Was macht der Computer?

Befehlsformat  
Assembler

Andere Darstellung  
des Befehles

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	s	s	0	0	1	0		2. Byte
									3. Byte
									LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Gesetzt, wenn Übertrag von Bit 12, sonst rückgesetzt
P/V	Gesetzt bei Überlauf, sonst rückgesetzt
N	Gesetzt
C	Gesetzt bei Übertrag, sonst rückgesetzt

## Kommentar

Der Inhalt des Registerpaares ss (BC, DE, HL, SP), sowie das CY-Bit wird vom Inhalt von HL subtrahiert.

Das Ergebnis steht dann in HL.

Registerpaar	BC	DE	HL	SP
ss	00	01	10	11

Beispiel: HL enthalte 9999H, BC enthalte 1111H, CY sei gesetzt.  
Nach der Ausführung des Befehls

SBC HL, BC

enthält HL 8887H.

# ADD IX, pp

Addition von IX und Registerpaar

IX ← IX+pp			
Marke	Opcode	Operanden	Kommentar
Label:	ADD	IX,pp	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	0	1	1	1	0	1

0	0	p	p	1	0	0	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

DD	MSB
----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Gesetzt bei Übertrag aus Bit 11, sonst rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Gesetzt bei Übertrag aus Bit 15, sonst rückgesetzt

## Kommentar

Der Inhalt des Registerpaares pp (BC, DE, IX, SP) wird zum Inhalt des Registers IX addiert.

Das Ergebnis steht anschließend in IX.

Registerpaar	BC	DE	IX	SP
pp	00	01	10	11

Beispiel: IX enthalte 333H, BC enthalte 5555H.  
Nach der Ausführung des Befehls

ADD IX,BC

enthält IX 5888H.

# ADD IY, rr

Addition von IY und Registerpaar

IY ← IY + rr			
Marke	Opcode	Operanden	Kommentar
Label:	ADD	IY,rr	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1
0	0	r	r	1	0	0	1

in Sedezimaler Form

FD	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Gesetzt bei Übertrag aus Bit 11, sonst rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Gesetzt bei Übertrag aus Bit 15, sonst rückgesetzt

## Kommentar

Der Inhalt des Registerpaars rr (BC, DE, IY, SP) wird zum Inhalt des Registers IY addiert.

Das Ergebnis steht anschließend in IY.

Registerpaar	BC	DE	IY	SP
rr	00	01	10	11

Beispiel: IY enthalte 333H, DE enthalte 555H.  
Nach Ausführung des Befehls

ADD IY,DE

enthält IY 888H.

# INC ss

Inkrementieren eines Registerpaars

SS ← SS + 1			
Marke	Opcode	Operanden	Kommentar
Label:	INC	ss	; comment

...macht der Computer?

...Format Assembler

...Darstellung Befehles

in Sedezimaler Form									
7	6	5	4	3	2	1	0		MSB
0	0	s	s	0	0	1	1		
									2. Byte
									3. Byte
									LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.5 $\mu$ s / 2.4 $\mu$ s	1	6

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

Zum Inhalt des Registerpaares ss (BC, DE, HL, SP) wird 1 addiert.

Registerpaar	BC	DE	HL	SP
ss	00	01	10	11

Beispiel: HL enthalte 1000H.  
Nach Ausführung des Befehls

INC HL

enthält HL 1001H.

# INC IX

Inkrementieren des Indexregisters

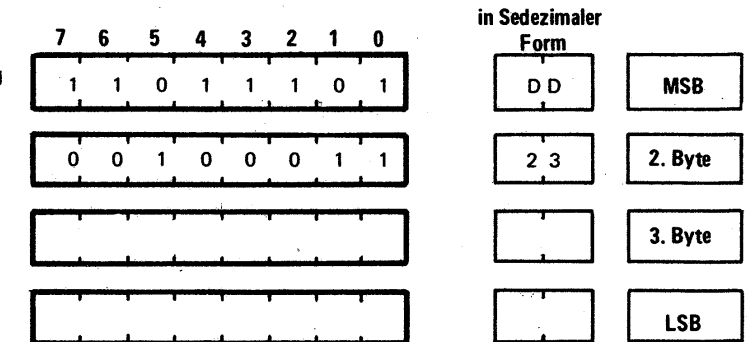
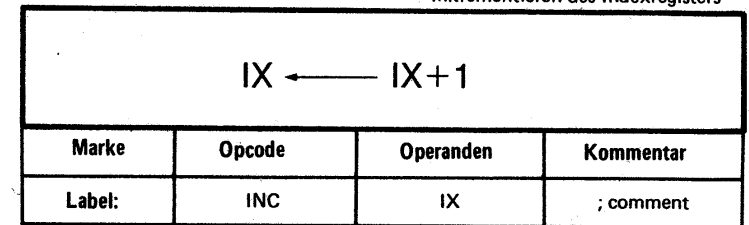
Was macht der Computer?

Defehlsformat in Assembler

Binäre Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflußt



Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	2	10 (4,6)

S	---
Z	---
H	---
P/V	---
N	---
C	---

**Kommentar**

Zum Inhalt des Registers IX wird 1 addiert.

Beispiel: IX enthalte 3300H.  
Nach Ausführung des Befehls

**INC IX**

enthält IX 3301H.

# INC IX

Inkrementieren des Indexregisters

IX ← IX+1			
Marke	Opcode	Operanden	Kommentar
Label:	INC	IX	; comment

... macht der Computer?

... Zahlensformat Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1
0	0	1	0	0	0	1	1

in Sedezimaler Form

FD	MSB
23	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	2	10 (4,6)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---



Kommentar

Zum Inhalt von IY wird 1 addiert.

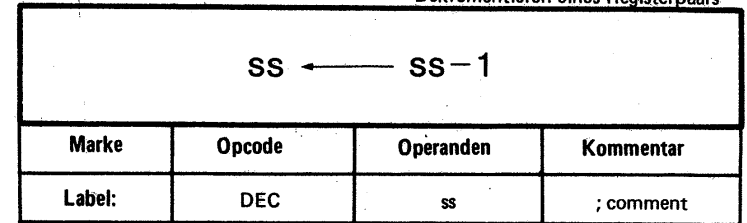
Beispiel: IY enthalte den Wert 2977H.  
Nach Ausführung des Befehls

INC IY

enthält IY den Wert 2978H.

# DEC ss

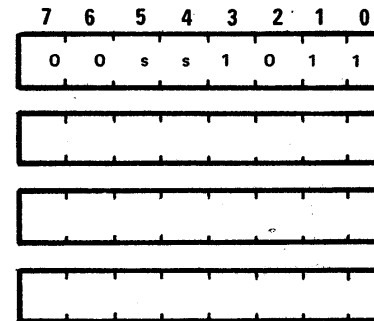
Dekrementieren eines Registerpaars



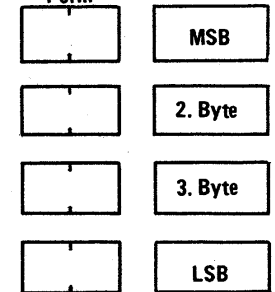
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1.5 $\mu\text{s}$ / 2.4 $\mu\text{s}$	1	6

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

### Kommentar

Vom Inhalt des Registerpaares ss (BC, DE, HL, SP) wird 1 abgezogen.

Registerpaar	BC	DE	HL	SP
ss	00	01	10	11

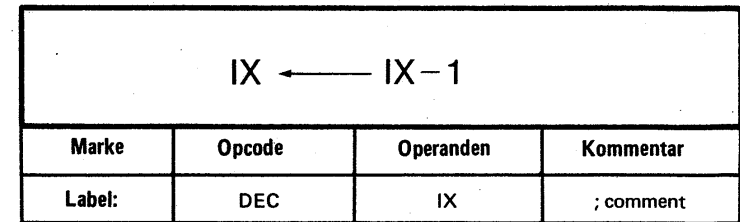
Beispiel: HL enthalte den Wert 1000H.  
Nach Ausführung des Befehls

DEC HL

enthält HL den Wert FFFH.

# DEC IX

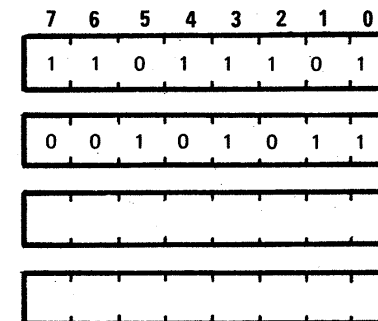
Dekrementieren eines Indexregisters



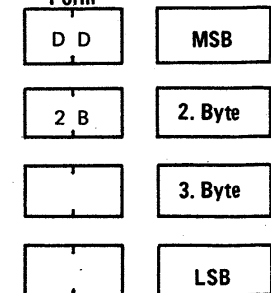
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	2	10 (4,6)

Folgende Status-Bits werden beeinflusst

S	
Z	
H	
P/V	
N	
C	

**Kommentar**

Vom Inhalt von IX wird 1 abgezogen.

Beispiel: IX enthalte den Wert 2006H.  
Nach der Ausführung des Befehls

**DEX IX**

enthält IX den Wert 2005H.

# DEC IX

Dekrementieren eines Indexregisters

$IY \leftarrow IY - 1$			
Marke	Opcode	Operanden	Kommentar
Label:	DEC	IY	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1
0	0	1	0	1	0	1	1

in Sedezimaler Form

FD	MSB
2B	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu\text{s}$ / 4 $\mu\text{s}$	2	10 (4,6)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

Kommentar

Vom Inhalt von IY wird 1 abgezogen.

Beispiel: IY enthalte den Wert 7649H.  
Nach Ausführung des Befehls

DEC IY

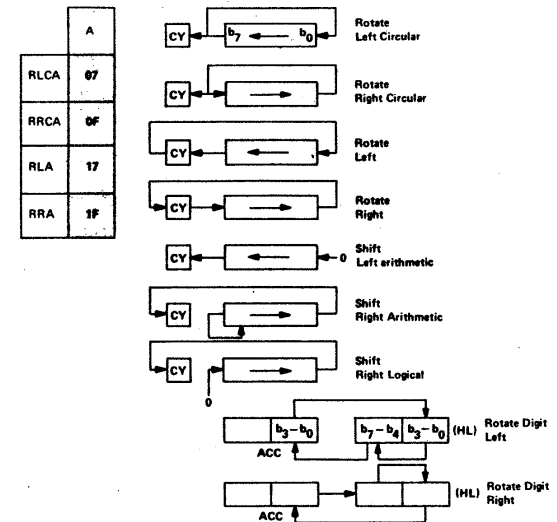
enthält IY den Wert 7648H.

# Rotations- und Schiebe-Befehle

Quelle und Ziel

	A	B	C	D	E	H	L	(HL)	(IX + d)	(IY + d)
'RLC'	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
'RRC'	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
'RL'	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
'RR'	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
'SLA'	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
'SRA'	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
'SRL'	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
'RLD'								ED 8F		
'RRD'								ED 87		

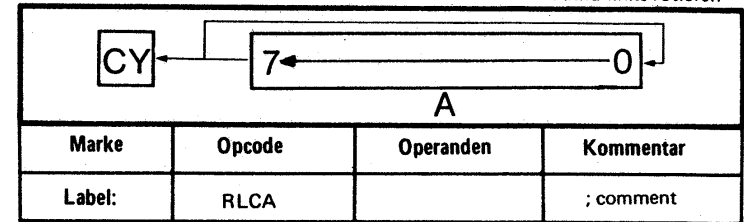
Operations-  
typ



# RLCA

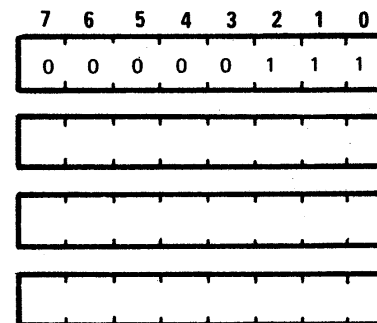
Akku links rotieren

Was macht der Computer?

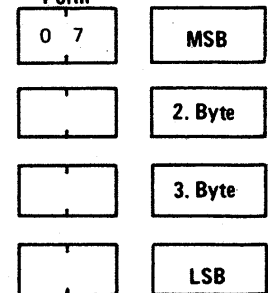


Befehlsformat  
in Assembler

Binäre Darstellung  
des Befehles



in Sedezimaler  
Form



Daten zur Befehls-  
ausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits  
werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	= Bit 7 des Akkumulators

## Kommentar

Der Inhalt des Akkumulators wird nach links rotiert. \*  
 Bit 0 kommt nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt von Bit 7 wird sowohl ins CY-Bit als auch in das Bit 0 transferiert.

Beispiel: A enthalte 1 0 0 0 1 0 0 0  
 Nach der Ausführung des Befehls

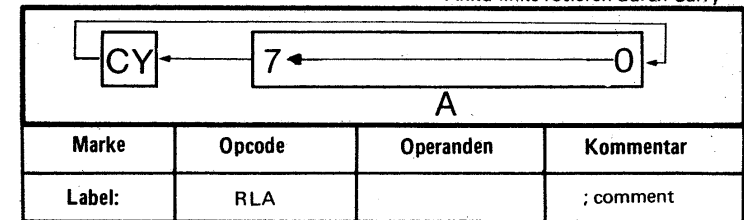
RLCA

enthält der Akkumulator 0 0 0 1 0 0 0 1 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition.

# RLA

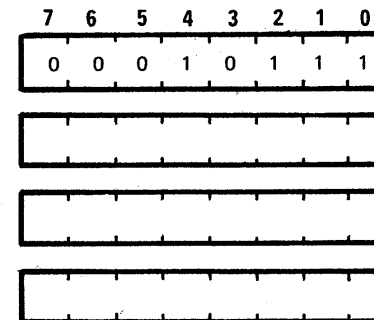
Akku links rotieren durch Carry



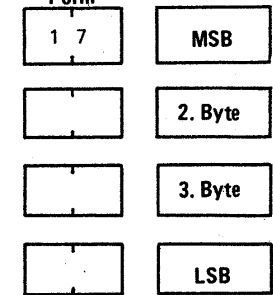
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Inhalt des Bit 7 von A

## Kommentar

Der Inhalt von A wird gemeinsam mit dem CY-Bit um 1 Bitposition nach links rotiert, Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Bit 7 kommt ins CY-Bit, während dessen Inhalt nach Bit 0 übertragen wird.

Beispiel: Durch die Ausführung des Befehls

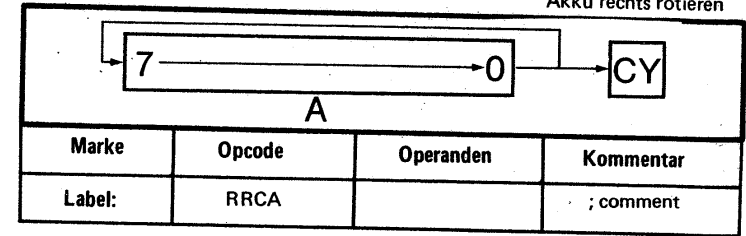
RLA

ergibt sich folgende Situation:

	CY	A
vor Ausführung des Befehls	1	0 1 1 1 0 1 1 0
nach Ausführung des Befehls	0	1 1 1 0 1 1 0 1

# RRCA

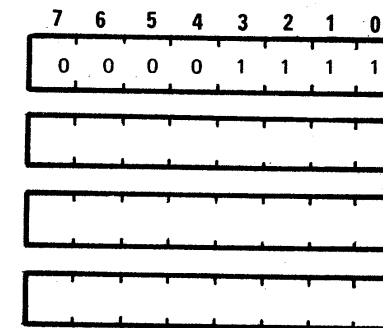
Akku rechts rotieren



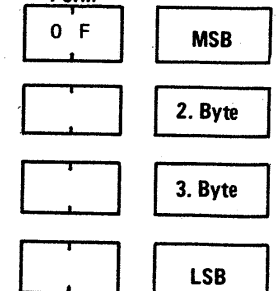
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Inhalt von Bit 0 von A

## Kommentar

Der Inhalt des Akkumulators wird nach rechts rotiert. \*  
Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Der Inhalt von Bit 0 wird sowohl nach Bit 7 als auch ins CY-Bit  
transferiert.

Beispiel: Der Akkumulator enthalte 0 0 0 1 0 0 0 1  
Nach der Ausführung des Befehls

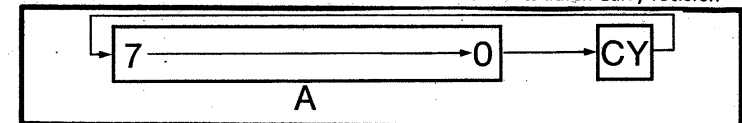
RRCA

enthält der Akkumulator 1 0 0 0 1 0 0 0 und CY 1

\* Bemerkung: Rotiert wird um eine Bitposition

# RRA

Akku rechts durch Carry rotieren



Marke	Opcode	Operanden	Kommentar
Label:	RRCA		; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	1	1	1	1	1

in Sedezimaler Form

1 F	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s/ 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Inhalt von Bit 0 von A



# Kommentar

Der Inhalt des Akkumulators wird gemeinsam mit dem CY-Bit um 1 Bitposition nach rechts rotiert, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 gelangt nach CY und der Inhalt von CY nach Bit 7.

Beispiel: Durch Ausführung des Befehls

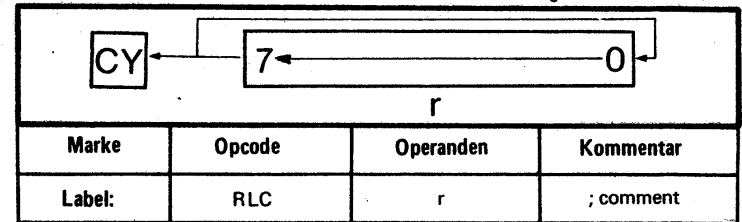
RRA

ergibt sich folgende Situation:

	A	CY
vor Ausführung des Befehls	1 1 1 0 0 0 0 1	0
nach Ausführung des Befehls	0 1 1 1 0 0 0 0	1

# RLC r

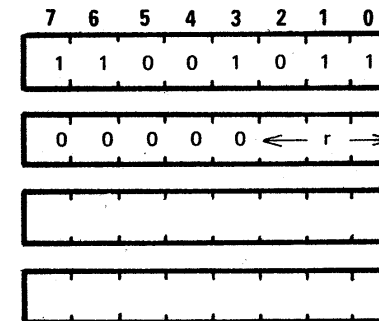
Register links rotieren



Wicht der Register?

Format Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

C B	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 7 von A

## Kommentar

Der Inhalt des Registers r wird nach links rotiert. \*  
 Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt von Bit 7 gelangt sowohl nach Bit 0 als auch ins CY-Bit.

Beispiel: Der Inhalt des Registers C sei 1 0 0 0 1 0 0 0  
 Nach der Ausführung des Befehls

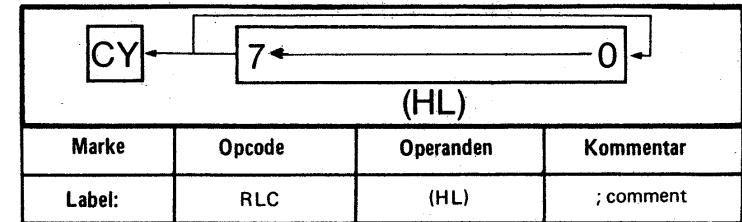
RLC C

enthält das Register C 0 0 0 1 0 0 0 1 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition.

# RLC [HL]

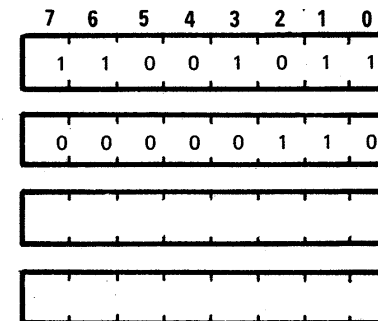
Speicherstelle links rotieren



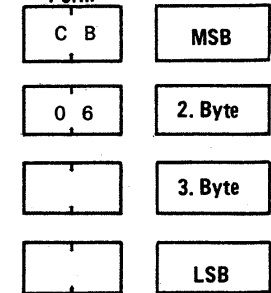
Was macht der Computer?

Befehlsformat  
 Assembler

Binäre Darstellung  
 des Befehles



in Sedezimaler  
 Form



Daten zur Befehls-  
 ausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s/ 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits  
 werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 7

Kommentar

Der Inhalt der Speicherstelle, deren Adresse durch den Inhalt des HL-Registerpaares angegeben wird, wird nach links rotiert. \*  
 Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt des Bit 7 gelangt sowohl nach Bit 0 als auch ins CY-Bit.

Beispiel: HL enthalte 2828H und die Speicherstelle 2828 enthalte 10001000.

Nach der Ausführung des Befehls

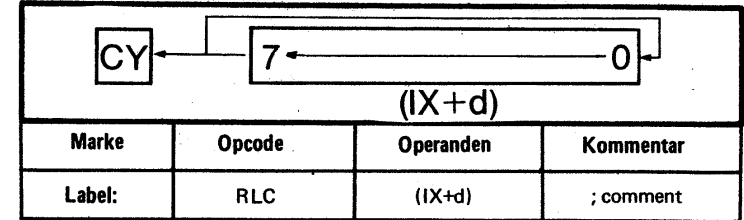
RLC (HL)

enthält die Speicherstelle 2828 00010001 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition.

# RLC [IX+d]

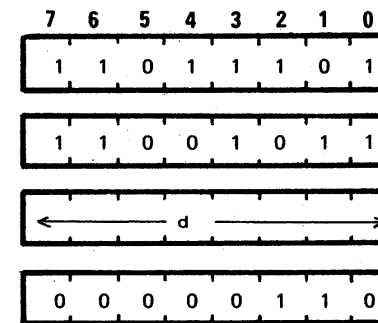
Indiziert adressierte Speicherstelle links rotieren



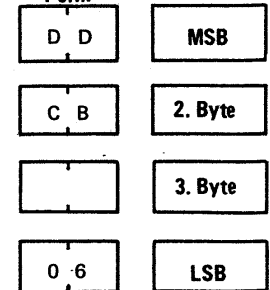
... macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt, wenn Parität gerade, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 7

## Kommentar

Der Inhalt des IX-Registers, addiert mit einer Zahl d (in 2-Komplement-Darstellung), wird als Speicheradresse interpretiert. Der Inhalt dieser Speicherstelle wird nach links rotiert. \*  
Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt von Bit 7 gelangt sowohl nach Bit 0 als auch nach CY.

Beispiel: Der Inhalt von IX sei 1000H, der Inhalt der Speicherstelle 1002 sei 1 0 0 0 1 0 0 0.

Nach der Ausführung des Befehls

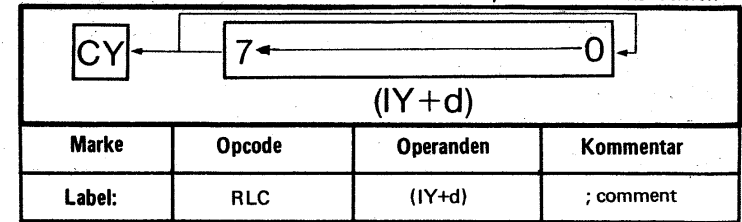
RLC (IX+2H)

enthält die Speicherstelle 1002 0 0 0 1 0 0 0 1 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition

# RLC [IY+d]

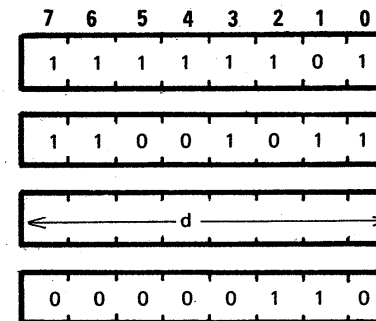
Indiziert adressierte Speicherstelle links rotieren



...macht der Computer?

...Format Assembler

...Darstellung des Befehls



in Sedezimaler Form

F D	MSB
C B	2. Byte
	3. Byte
0 6	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 7

## Kommentar

Der Inhalt des IY-Registers, addiert mit einer Zahl d (in 2-Komplement-Darstellung) wird als Speicheradresse interpretiert. Der Inhalt dieser Speicherstelle wird nach links rotiert. \*  
 Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt von Bit 7 gelangt sowohl nach Bit 0 als auch nach CY.

Beispiel: Der Inhalt von IY sei 1000H, der Inhalt der Speicherstelle 1002 sei 1 0 0 0 1 0 0 0.  
 Nach Ausführung des Befehls

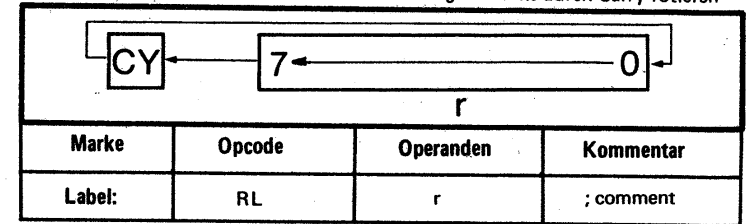
RLC (IY+2H)

enthält die Speicherstelle 1002 0 0 0 1 0 0 0 1 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition.

# RL r

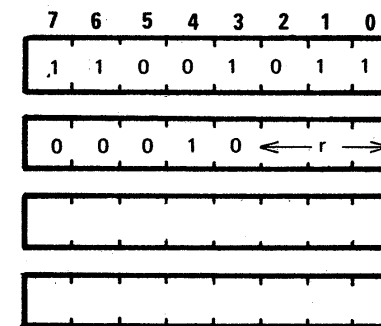
Register links durch Carry rotieren



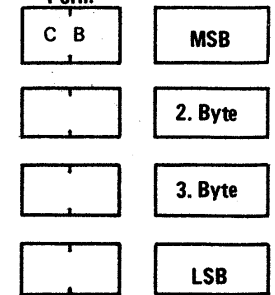
Wird der Register?

Format Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 7

## Kommentar

Der Inhalt des Operanden wird mit dem CY-Bit nach links rotiert. \*  
 Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Bit 7 gelangt ins CY-Bit, während dieses nach Bit 0 transferiert wird.

Beispiel: Der Inhalt von D sei 1 0 0 1 1 1 1 und CY sei 1.  
 Nach Ausführung des Befehls

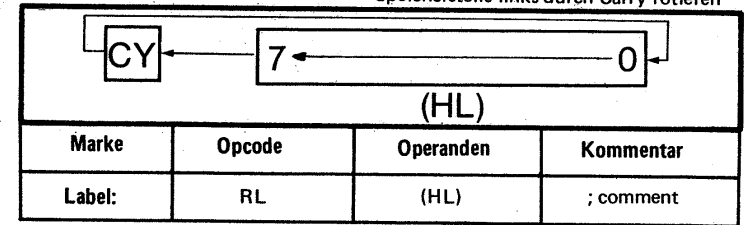
RL D

enthält D 0 0 0 1 1 1 1 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition.

# RL [HL]

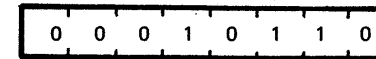
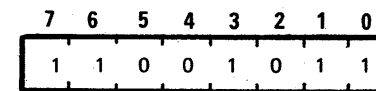
Speicherstelle links durch Carry rotieren



Wicht der  
 Operater?

Format  
 Assembler

in Darstellung  
 Befehles



in Sedezimaler  
 Form



daten zur Befehls-  
 Ausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Ma- schinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu\text{s}$ / 6 $\mu\text{s}$	4	15 (4,4,4,3)

Folgende Status-Bits  
 werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 7

## Kommentar

Der Inhalt des Operanden wird mit dem CY-Bit nach links rotiert. Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw.

Bit 7 gelangt ins CY-Bit, während dieses nach Bit 0 transferiert wird.

Beispiel: HL enthalte 2525H, die Speicherstelle 2525H enthalte 1 0 0 0 1 1 1 1 und CY sei 0.  
Nach der Ausführung des Befehls

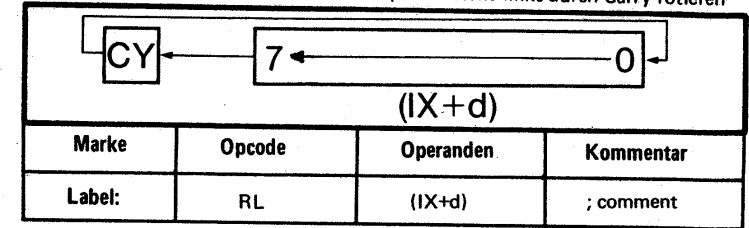
RL (HL)

enthält die Speicherstelle 2525H 0 0 0 1 1 1 1 0 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition

# RL [IX+d]

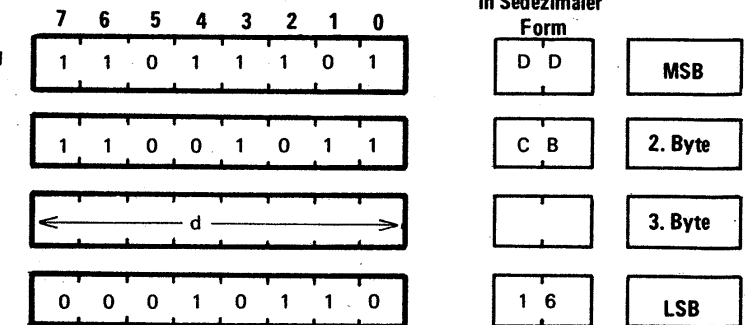
Indiziert adressierte Speicherstelle links durch Carry rotieren



Was macht der Computer?

Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität
N	Rückgesetzt
C	Inhalt von Bit 7

## Kommentar

Der Inhalt von IX, addiert mit der Zahl d (in 2-Komplement-Darstellung), wird als Speicheradresse interpretiert.

Der Inhalt des Operanden wird mit dem CY-Bit nach links rotiert.

Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw. Bit 7 gelangt ins CY-Bit, während dieses nach Bit 0 transferiert wird.

Beispiel: IX enthalte den Wert 2525H, die Speicherstelle 2528H den Wert 1 0 0 0 1 1 1 1 und CY sei rückgesetzt (= 0). Nach Ausführung des Befehls

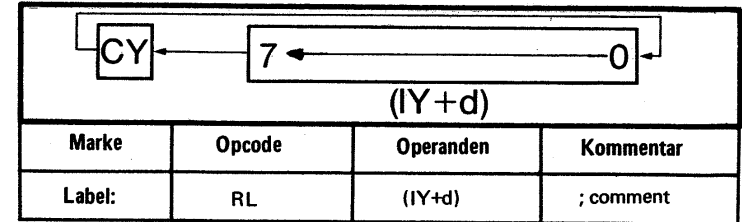
RL (IX+3)

enthält die Speicherstelle 2528H den Wert 0 0 0 1 1 1 1 0 und das Carrybit CY ist gesetzt (= 1).

\* Bemerkung: Rotiert wird um eine Bitposition.

# RL [IX+d]

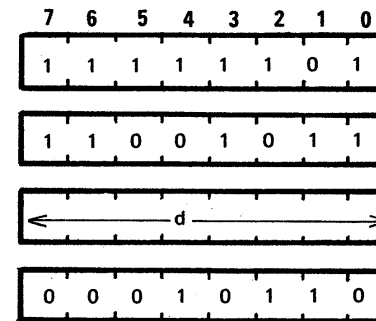
Indiziert adressierte Speicherstelle links durch Carry rotieren



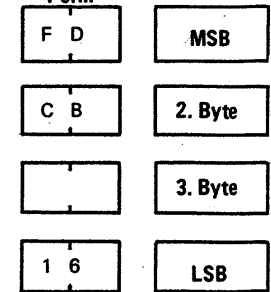
Was macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5,75 $\mu$ s/ 9,2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 7



## Kommentar

Der Inhalt von IY, addiert mit der Zahl d (in 2-Komplement-Darstellung), wird als Speicheradresse interpretiert.

Der Inhalt des Operanden wird mit dem CY-Bit nach links rotiert.

Bit 0 nach Bit 1, Bit 1 nach Bit 2 usw. Bit 7 gelangt ins CY-Bit, während dieses nach Bit 0 transferiert wird.

Beispiel: IY enthalte den Wert 2525H, die Speicherstelle 2528H den Wert 10001111 und CY sei rückgesetzt (= 0).  
Nach Ausführung des Befehls

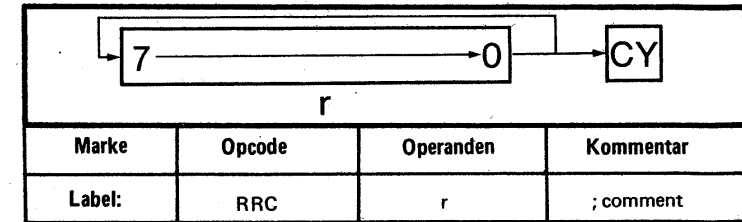
RL (IY+3)

enthält die Speicherstelle 2528H den Wert 00011110 und das Carrybit CY ist gesetzt (= 1).

\* Bemerkung: Rotiert wird um eine Bitposition.

# RRC r

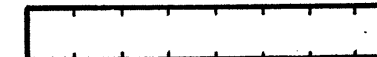
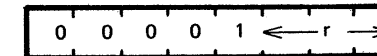
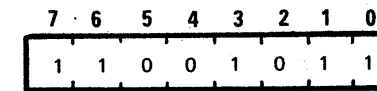
Register rechts rotieren



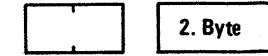
Was macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des Operanden wird rechts rotiert, \* der Inhalt von Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 wird sowohl nach Bit 0 als auch nach CY transferiert.

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

Beispiel: Register L enthalte 00110001.  
Nach der Ausführung des Befehls

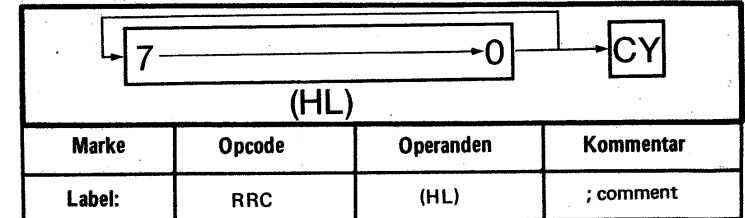
RRC L

enthält L 10011000 und CY 1.

\* Bemerkung: Rotiert wird um eine Bitposition

# RRC [HL]

Speicherstelle rechts rotieren



Was macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	1

0	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

C B	MSB
-----	-----

0 E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt, wenn gerade Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des HL-Registerpaares wird als Speicheradresse interpretiert.  
Der Inhalt dieser Zelle wird um 1 Bitposition nach rechts rotiert.

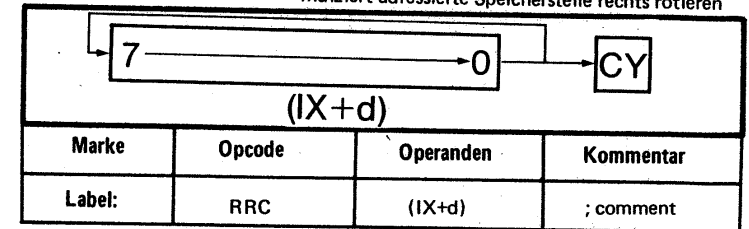
Beispiel: Das Registerpaar HL enthalte den Wert 4711.  
In Speicherstelle 4711H stehe die Binärzahl 00011000,  
das Übertragungsbedingungsbit (CY-Flag) des Registers F  
sei gesetzt (= 1).  
Nach Ausführung des Befehls

RRC (HL)

steht in Speicherstelle 4711H die Binärzahl 00001100  
und das Übertragungsbedingungsbit CY ist rückgesetzt (= 0).

# RRC [IX+d]

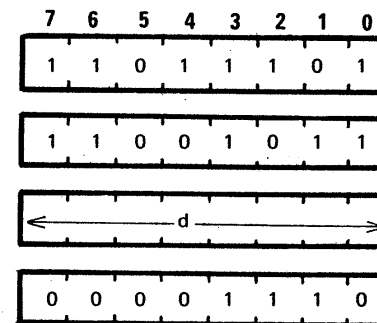
Indiziert adressierte Speicherstelle rechts rotieren



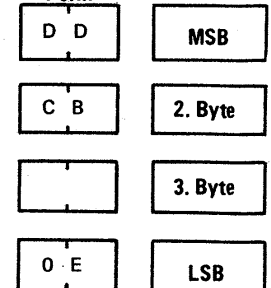
Was macht der Computer?

Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt von IX, addiert mit d (im 2-Komplement) wird als Speicheradresse interpretiert, der Inhalt dieser Zelle nach rechts rotiert.\*

Beispiel: Das Doppelregister IX enthalte den Wert 4711H.  
 In Speicherstelle 4715H stehe die Binärzahl 00011000, das Übertragungsbit (CY-Flag) des Registers F sei gesetzt (= 1).  
 Nach Ausführung des Befehls

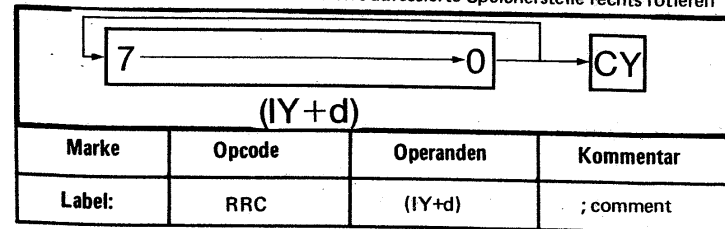
RRC (IX+4)

steht in Speicherstelle 4715H die Binärzahl 00001100 und das Übertragungsbit CY ist rückgesetzt (= 0).

Bemerkung: Rotiert wird um 1 Bitposition.

# RRC [IX+d]

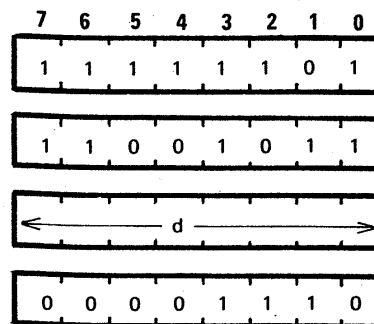
Indiziert adressierte Speicherstelle rechts rotieren



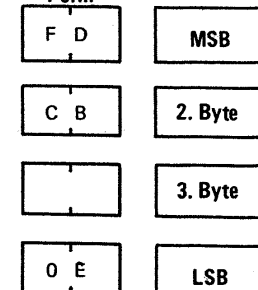
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement) wird als Speicheradresse interpretiert, der Inhalt dieser Zelle um 1 Bitposition nach rechts rotiert.

Beispiel: Das Doppelregister IY enthalte den Wert 4711H. In Speicherstelle 4715H stehe die Binärzahl 0 0 0 1 1 0 0 0, das Übertragungsbedingungsbit (CY-Flag) des Registers F sei gesetzt (= 1).

Nach der Ausführung des Befehls

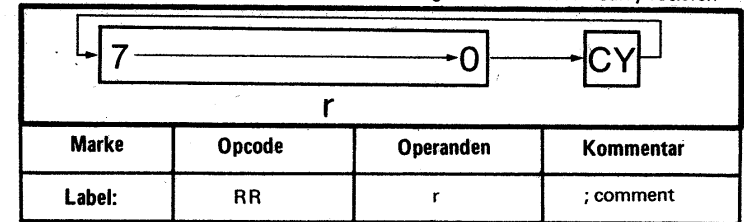
RRC (IY+d)

steht in Speicherstelle 4715H die Binärzahl 0 0 0 0 1 1 0 0 und das Übertragsbit CY ist rückgesetzt (= 0).

# RR r

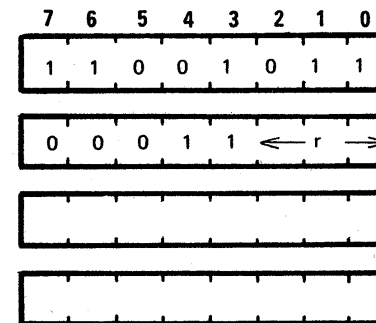
Register rechts durch Carry rotieren

Was macht der Computer?

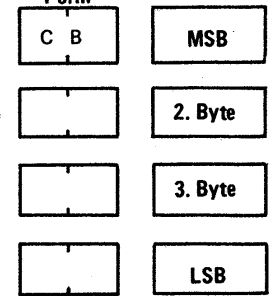


Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu\text{s}$ / 3.2 $\mu\text{s}$	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des Operanden wird gemeinsam mit CY nach rechts rotiert.\* Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw. Bit 0 gelangt ins CY-Bit, dieses wird nach Bit 7 transferiert.

Die für den Befehl zulässigen Registernamen werden im Maschinen-code in folgender Weise codiert:

Registername	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

Beispiel: Das Register D enthalte 1 1 0 0 0 1 0 0 . CY enthalte 0.  
Nach der Ausführung des Befehls

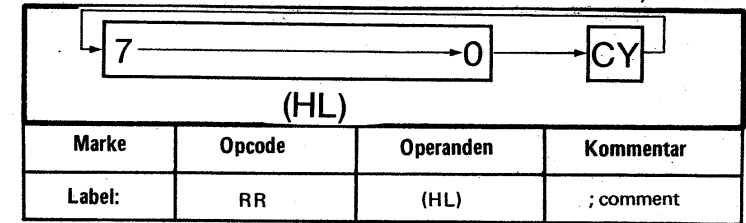
RR D

enthält D 0 1 1 0 0 0 1 0 und CY 0.

\* Bemerkung: Rotiert wird um 1 Bitposition

# RR [HL]

Speicherstelle rechts durch Carry rotieren



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	0	0	1	0	1	1	C B	MSB
0	0	0	1	1	1	1	0	1 E	2. Byte
									3. Byte
									LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s/ 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des HL-Registerpaares wird als Speicheradresse interpretiert, der Inhalt dieser Zelle gemeinsam mit CY um 1 Bitposition nach rechts verschoben.

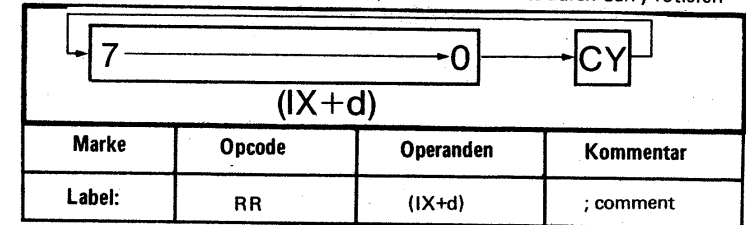
Beispiel: Das Registerpaar HL enthalte den Wert 4711H, die Speicherstelle 4711H den Binärwert 1 1 0 0 0 1 0 0 .  
Das Übertragungsbit CY sei rückgesetzt (= 0).  
Nach der Ausführung des Befehls

RR (HL)

enthält die Speicherstelle 4711H den Binärwert 0 1 1 0 0 0 1 0 und das CY-Bit ist rückgesetzt (= 0).

# RR [IX+d]

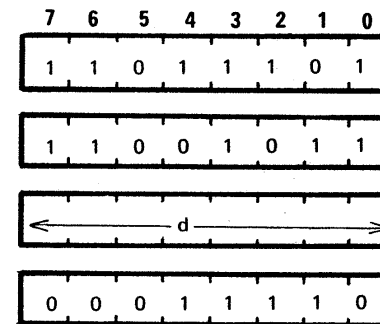
Indiziert adressierte Speicherstelle rechts durch Carry rotieren



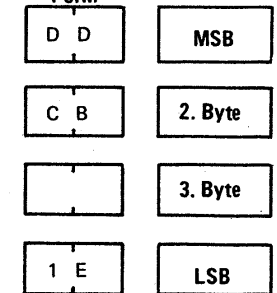
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu\text{s}$ / 9.2 $\mu\text{s}$	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt, wenn Parität gerade, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt von IX, addiert mit d (im 2-Komplement) wird als Speicheradresse interpretiert, der Inhalt dieser Zelle gemeinsam mit CY um 1 Bitposition nach rechts verschoben.

Beispiel: Das Doppelregister IX enthalte den Wert 4711H, in Speicherstelle 4715H stehe die Binärzahl 1 1 0 0 0 1 0 0, das Übertragsbedingungsbit (CY-Flag) des Registers F sei gesetzt (= 1).

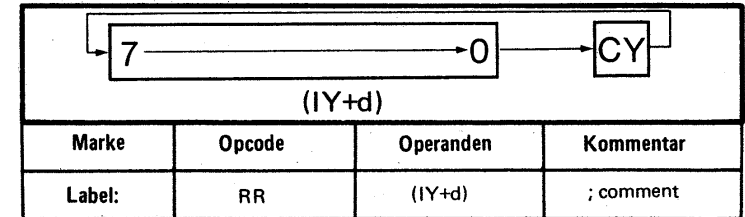
Nach der Ausführung des Befehls

RR (IX+4)

steht in Speicherstelle 4715H die Binärzahl 0 1 1 0 0 0 1 0 und das Übertragsbit CY ist rückgesetzt (= 0).

# RR [IY+d]

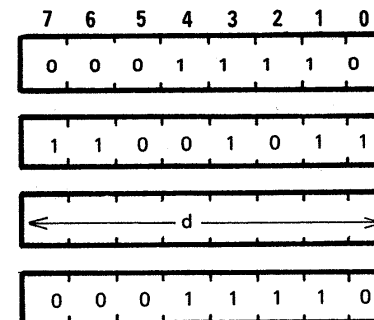
Indiziert adressierte Speicherstelle durch Carry rotieren



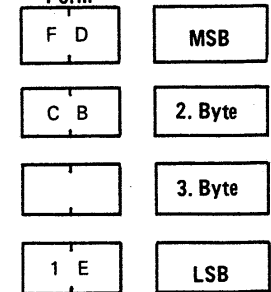
Was macht der Computer?

Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0



## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement) wird als Speicheradresse interpretiert, der Inhalt dieser Zelle gemeinsam mit CY um 1 Bitposition nach rechts verschoben.

Beispiel: Das Doppelregister IY enthalte den Wert 4711H, in Speicherstelle 4715H stehe die Binärzahl 11000100, das Übertragsbedingungsbit (CY-Flag) des Registers F sei gesetzt (= 1).

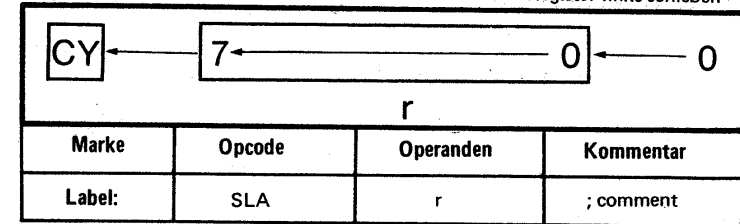
Nach Ausführung des Befehls

RR (IY+4)

steht in Speicherstelle 4715H die Binärzahl 01100010 und das Übertragsbit CY ist rückgesetzt (= 0).

# SLA r

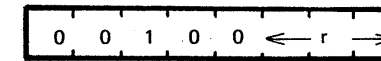
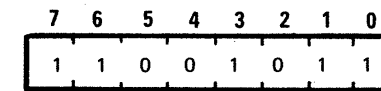
Register links schieben



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	= Inhalt von Bit 7

## Kommentar

Der Inhalt des Operanden wird nach links geschoben;\* dabei wird Bit 0 rückgesetzt, Bit 0 kommt nach Bit 1, Bit 1 nach Bit 2 usw.

Der Inhalt von Bit 7 wird ins CY-Bit abgebildet.

Die für den Befehl zulässigen Registernamen werden im Maschinen-code in folgender Weise codiert:

Registername	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

Beispiel: Der Inhalt von H sei 00100110.  
Nach der Ausführung des Befehls

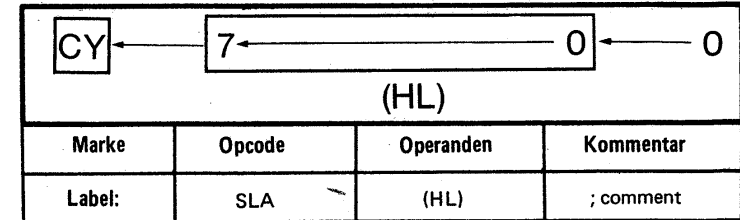
SLA H

enthält H 01001100 und CY 0 unabhängig vom bisherigen Wert des CY-Bits.

\* Bemerkung: Geschoben wird um 1 Bitposition

# SLA [HL]

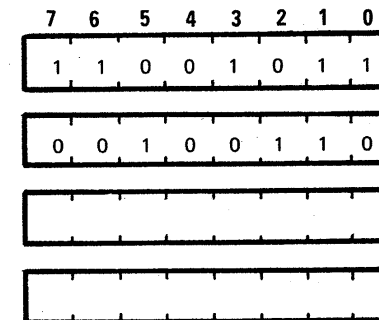
Speicherstelle links schieben



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

C B	MSB
2 6	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität
N	Rückgesetzt
C	Inhalt von Bit 7

## Kommentar

Der Inhalt von HL wird als Speicheradresse interpretiert, deren Inhalt wird arithmetisch um 1 Bitposition nach links verschoben.

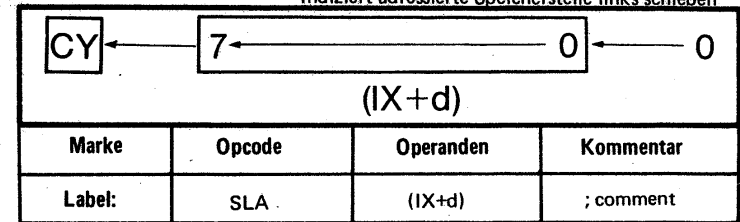
Beispiel: Der Inhalt von HL sei 4711H, der Inhalt von Speicherzelle 4711H sei 00100110.  
Nach der Ausführung des Befehls

SLA (HL)

enthält Speicherzelle 4711H den Wert 01001100 und das CY-Bit ist rückgesetzt (= 0), unabhängig von seinem bisherigen Wert.

# SLA [IX+d]

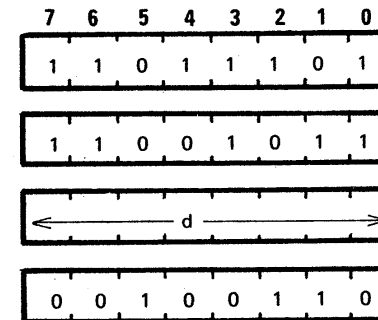
Indiziert adressierte Speicherstelle links schieben



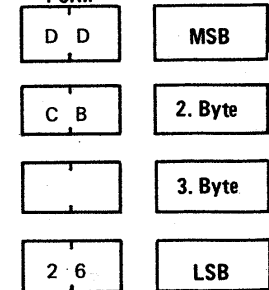
Wacht der Computer?

Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M)-Zyklen	Anzahl der Takt(=T)-Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 7

## Kommentar

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert, deren Inhalt arithmetisch nach links verschoben wird.\*

Beispiel: Der Inhalt von IX sei 4711H, der Inhalt der Speicherzelle 4713H sei 00100110.  
Nach der Ausführung des Befehls

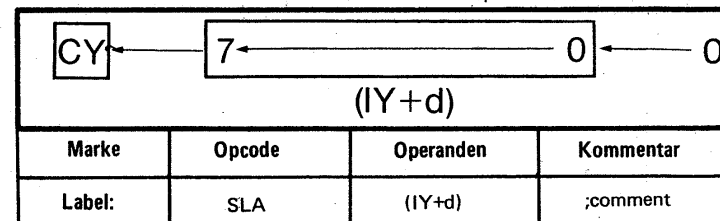
SLA (IX+2)

enthält Speicherzelle 4713H den Wert 01001100 und das CY-Bit ist rückgesetzt (= 0), unabhängig von seinem bisherigen Wert.

\* Bemerkung: Geschoben wird um 1 Bitposition.

# SLA [IY+d]

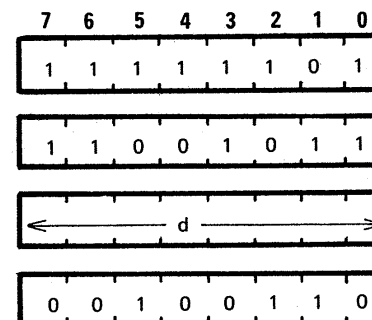
Indiziert adressierte Speicherstelle links schieben



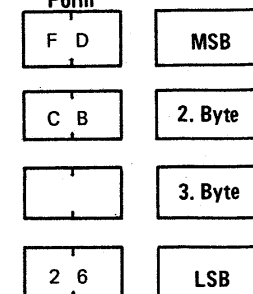
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 7

## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert, der Inhalt wird arithmetisch nach links verschoben.\*

Beispiel: Der Inhalt von IY sei 4711H, der Inhalt der Speicherzelle 4713H sei 0 0 1 0 0 1 1 0 .  
Nach der Ausführung des Befehls

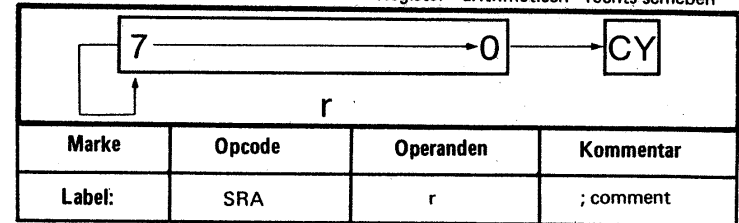
SLA (IY+2)

enthält Speicherzelle 4713H den Wert 0 1 0 0 1 1 0 0 und das CY-Bit ist rückgesetzt (= 0), unabhängig von seinem bisherigen Wert.

\* Bemerkung: Geschoben wird um 1 Bitposition.

# SRA r

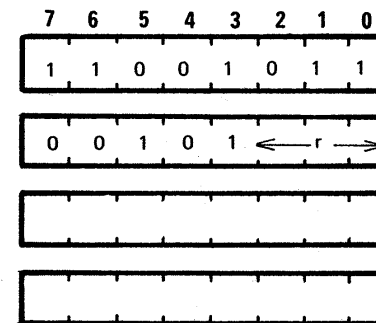
Register "arithmetisch" rechts schieben



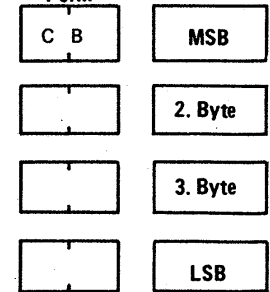
... macht der Computer?

Befehlsformat Assembler

binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des Operanden wird arithmetisch um 1 Bitposition nach rechts verschoben, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 wird in das CY-Bit transferiert, während der Inhalt von Bit 7 unverändert bleibt.

Die für den Befehl zulässigen Registernamen werden im Maschinencode in folgender Weise codiert:

Registername	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

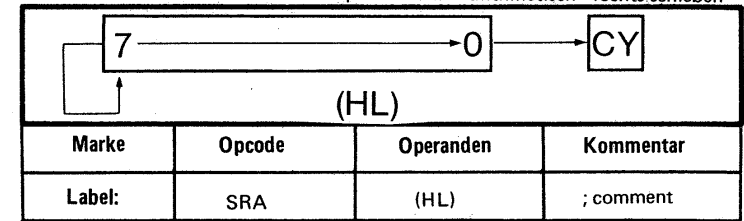
Beispiel: Register L enthalte 1 0 1 1 1 0 0 0 .  
Nach dem Befehl

SRA L

enthält Register L 1 1 0 1 1 1 0 0 und CY 0

# SRA [HL]

Speicherstelle "arithmetisch" rechts schieben



macht der  
puter?

hlsformat  
Assembler

äre Darstellung  
Befehles

7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	1

0	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler  
Form

C B	MSB
-----	-----

2 E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehls-  
ausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Ma- schinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu\text{s}$ / 6 $\mu\text{s}$	4	15 (4,4,4,3)

Folgende Status-Bits  
werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt des Bit 0

## Kommentar

Der Inhalt des Registerpaares HL wird als Speicheradresse interpretiert. Der Inhalt des Operanden wird arithmetisch um 1 Bitposition nach rechts verschoben, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 wird in das CY-Bit transferiert, während der Inhalt von Bit 7 unverändert bleibt.

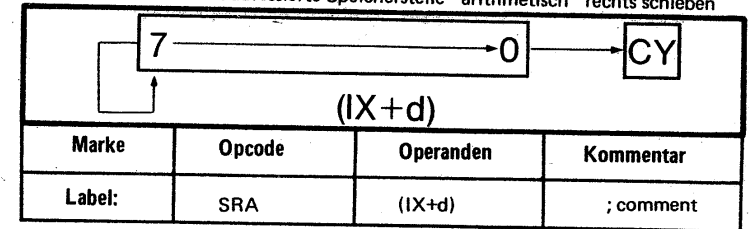
Beispiel: Das Registerpaar HL enthalte den Wert 0H, die Speicherstelle 0H den Binärwert 1 0 1 1 1 0 0 .  
Nach der Ausführung des Befehls

SRA (HL)

enthält die Speicherstelle 0H den Wert 1 1 0 1 1 1 0 0 und das CY-Bit ist (unabhängig von seinem bisherigen Wert) rückgesetzt.

# SRA [IX+d]

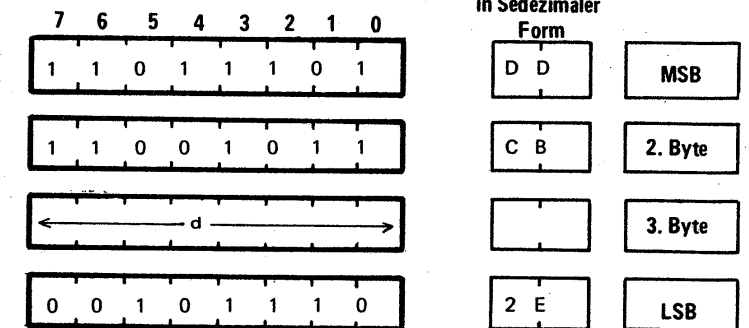
Indiziert adressierte Speicherstelle "arithmetisch" rechts schieben



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert. Der Inhalt des Operanden wird arithmetisch um 1 Bitposition nach rechts verschoben, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 wird in das CY-Bit transferiert, während der Inhalt von Bit 7 unverändert bleibt.

Beispiel: Das Registerpaar IX enthalte den Wert 0H, die Speicherstelle 0FH den Binärwert 1 0 1 1 1 0 0 0 .  
Nach der Ausführung des Befehls

SRA (IX+0FH)

enthält die Speicherstelle 0FH den Wert 1 1 0 1 1 1 0 0 und das CY-Bit ist (unabhängig von seinem bisherigen Wert) rückgesetzt.

# SRA [IX+d]

Was macht der Computer?

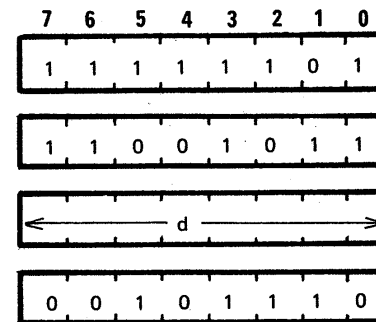
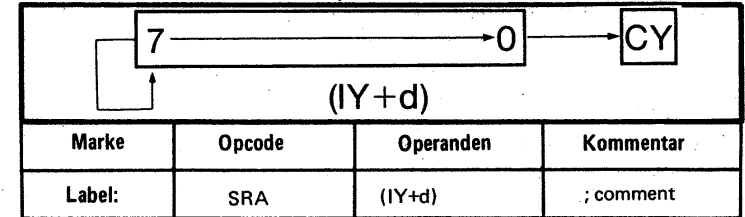
Befehlsformat in Assembler

Binäre Darstellung des Befehls

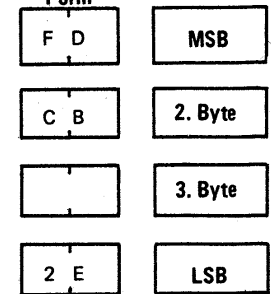
Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Indiziert adressierte Speicherstelle "arithmetisch" rechts schieben



in Sedezimaler Form



Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0



## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert. Der Inhalt des Operanden wird arithmetisch um 1 Bitposition nach rechts verschoben, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 0 wird in das CY-Bit transferiert, während der Inhalt von Bit 7 unverändert bleibt.

Beispiel: Das Registerpaar IY enthalte den Wert 0H, die Speicherstelle 0FH den Binärwert 1 0 1 1 1 0 0 0 .  
Nach der Ausführung des Befehls

SRA (IY+0FH)

enthält die Speicherstelle 0FH den Wert 1 1 0 1 1 1 0 0 und das CY-Bit ist (unabhängig von seinem bisherigen Wert) rückgesetzt.

# SRL r

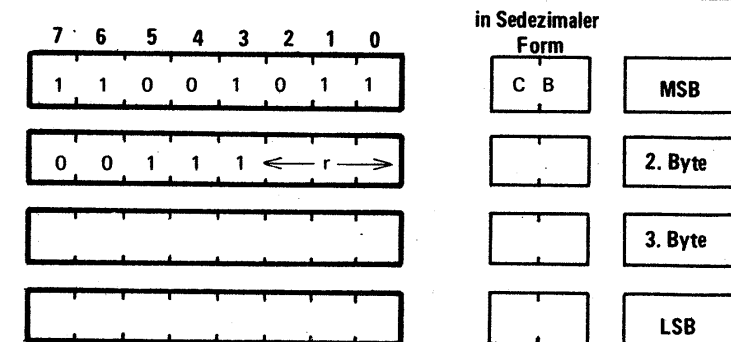
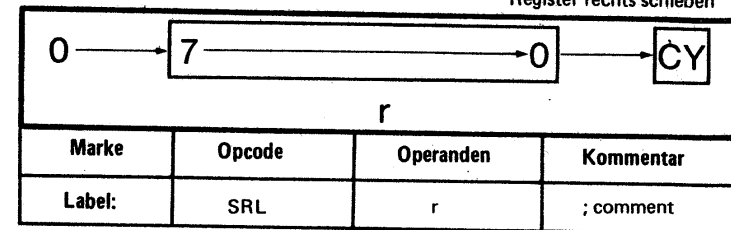
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

Folgende Status-Bits werden beeinflusst

Register rechts schieben



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt des Operanden wird um 1 Bitposition nach rechts verschoben, Bit 7 nach Bit 6, Bit 6 nach Bit 5 usw.

Bit 7 wird rückgesetzt, Bit 0 in CY transferiert.

Die für den Befehl zulässigen Registernamen werden im Maschinencode in folgender Weise codiert:

Registername	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

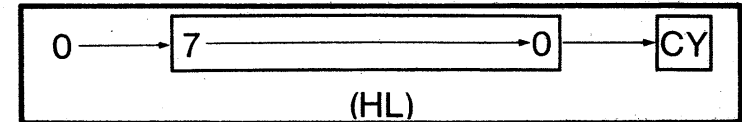
Beispiel: Das Register B enthalte den Binärwert 0 0 0 1 1 1 0 0  
Nach der Ausführung des Befehls

SRL B

enthält Register B den Wert 0 0 0 0 1 1 1 0 und CY 0.

# SRL [HL]

Speicherstelle rechts schieben



Marke	Opcode	Operanden	Kommentar
Label:	SRL	(HL)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	1

0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

C B	MSB
-----	-----

3 E	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt von HL wird als Speicheradresse interpretiert, deren Inhalt um 1 Bitposition nach rechts verschoben wird.

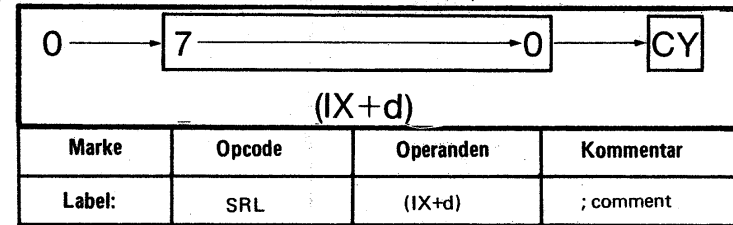
Beispiel: Das Registerpaar HL enthalte den Wert OFEH, die Speicherstelle OFEH die Binärzahl 00011100.  
Nach der Ausführung des Befehls

SRL (HL)

enthält die Speicherstelle OFEH den Wert 00001110 und das Übertragsbedingungsbit CY des Registers F ist unabhängig von seinem bisherigen Inhalt) rückgesetzt.

# SRL [IX+d]

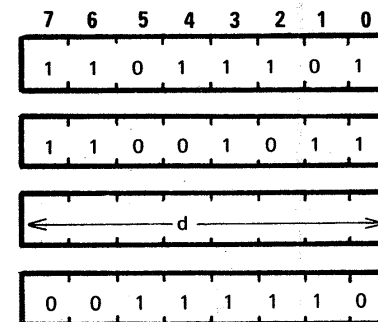
Indiziert adressierte Speicherstelle rechts schieben



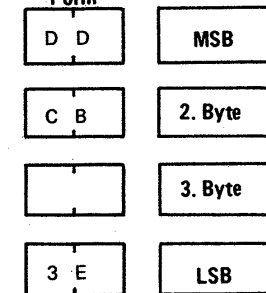
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

## Kommentar

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert, deren Inhalt um 1 Bitposition nach rechts verschoben wird.

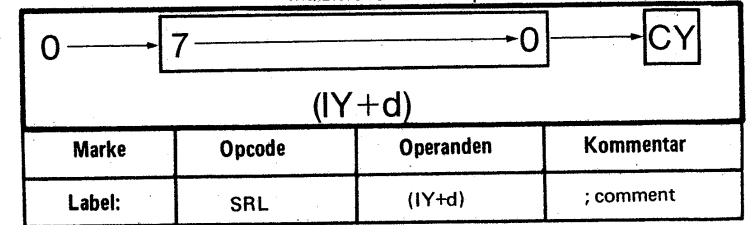
Beispiel: Das Registerpaar IX enthalte den Wert 0FEH, die Speicherstelle 0FFH die Binärzahl 00011100.  
Nach der Ausführung des Befehls

SRL (IX+1)

enthält die Speicherstelle 0FFH den Wert 00001110 und das Übertragsbedingungsbit CY des Registers F ist (unabhängig von seinem bisherigen Inhalt) rückgesetzt.

# SRL [IY+d]

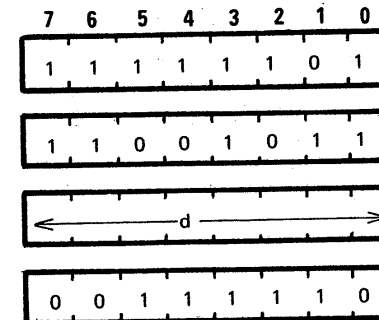
Indiziert adressierte Speicherstelle rechts schieben



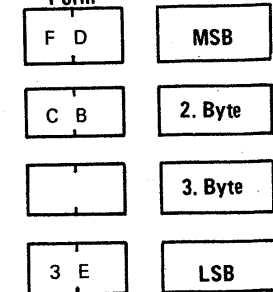
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Inhalt von Bit 0

Der Inhalt von IY, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert, deren Inhalt um 1 Bitposition nach rechts verschoben wird.

Beispiel: Das Registerpaar IY enthalte den Wert OFEH, die Speicherstelle OFFH die Binärzahl 0 0 0 1 1 1 0 0 .  
Nach der Ausführung des Befehls

SRL (IY+1)

enthält die Speicherstelle OFFH den Wert 0 0 0 0 1 1 1 0 und das Übertragsbedingungsbit CY des Registers F ist (unabhängig von seinem bisherigen Inhalt) rückgesetzt.

Was macht der Computer?

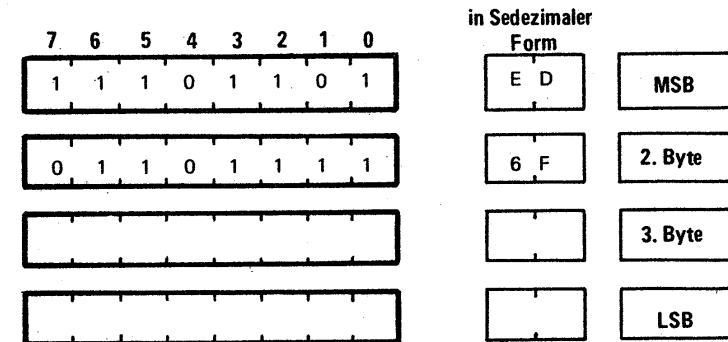
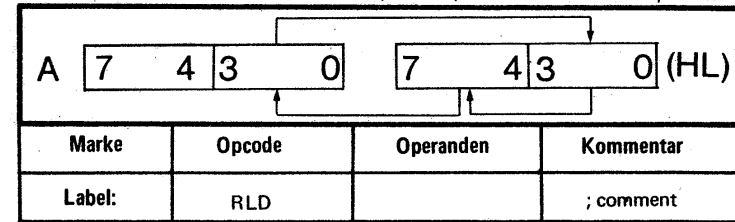
Befehlsformat in Assembler

Binäre Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Halbytes-Swap zwischen Akku und Speicher



Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.5 $\mu$ s / 7.2 $\mu$ s	5	18 (4,4,3,4,3)

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des HL-Registerpaares wird als Speicheradresse interpretiert.

Die unteren 4 Bits werden in die oberen 4 Bits übertragen, diese ihrerseits in die unteren 4 Bits des Akkumulators. Die unteren 4 Bits des Akkumulators werden in die unteren 4 Bits der Speicherzelle transferiert.

Beispiel: HL enthalte 5000H, in A sei 0 1 1 1 1 0 1 0 in 500H 0 0 1 1 0 0 0 1 .

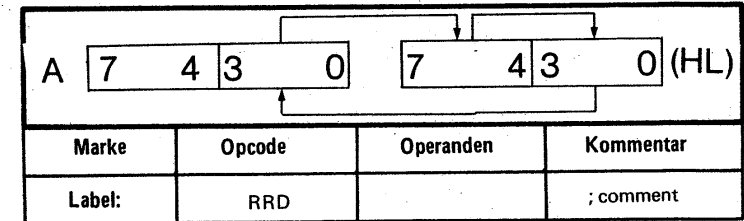
Nach der Ausführung des Befehls

RLD

enthält A 0 1 1 1 0 0 1 1 und 5000H 0 0 0 1 1 0 1 0..

# RRD

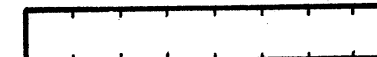
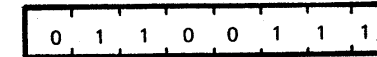
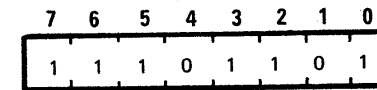
Halbbytes-Swap zwischen Akku und Speicher



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T)-Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.5 $\mu$ s / 7.2 $\mu$ s	5	18 (4,4,3,4,3)

Folgende Status-Bits werden beeinflusst

S	Gesetzt, wenn Ergebnis negativ, sonst rückgesetzt
Z	Gesetzt, wenn Ergebnis Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt, wenn Parität gerade, sonst rückgesetzt
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt von HL wird als Speicheradresse interpretiert, deren untere 4 Bits in die unteren 4 Bits des Akkumulators übertragen werden. Die unteren 4 Bits von A werden in die oberen 4 Bits der Speicherzelle, diese ihrerseits in die unteren 4 Bits transferiert.

Beispiel: HL enthalte 6000H, A enthalte 1 0 0 0 0 1 0 0 6000H  
 enthalte 0 0 1 0 0 0 0 0 .  
 Nach der Ausführung des Befehls

RRD

enthält A 1 0 0 0 0 0 0 0 und 6000H 0 1 0 0 0 0 1 0 .

# Befehle zur Einzelbit-

## Verarbeitung

BIT	REGISTER ADDRESSING							REG. INDIR.	INDEXED		
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
TEST 'BIT'	0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 46	FD CB d 46
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
	2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
	4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
	6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RESET BIT 'RES'	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE

SET BIT 'SET'	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6*
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

# BIT b, r

Bit in Register testen

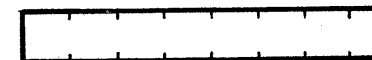
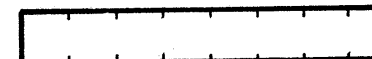
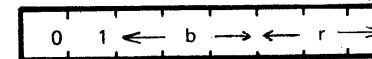
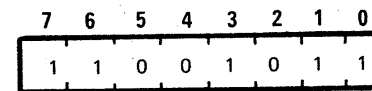
Was macht der Computer?

$$Z \leftarrow \bar{r}_b$$

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	BIT	b,r	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

C B	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Undefiniert
Z	Gesetzt, wenn Bit b = 0, sonst rückgesetzt
H	Gesetzt
P/V	Undefiniert
N	Rückgesetzt
C	Nicht beeinflusst



## Kommentar

Nach der Befehlsausführung enthält das Z-Bit das Komplement des getesteten Bits.

Getestetes Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

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

Beispiel: Bit 2 des Registers B enthalte 0.  
Nach der Ausführung des Befehls

BIT 2,B

enthält das Z-Bit 1, unabhängig von seinem bisherigen Zustand. Der Inhalt des Registers B bleibt unverändert.

# BIT b, [HL]

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Bit in Speicherstelle testen

Z ← (HL) <sub>b</sub>			
Marke	Opcode	Operanden	Kommentar
Label:	BIT	b, (HL)	; comment

7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	1

0	1	← b →	1	1	0
---	---	-------	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

C	B	MSB
---	---	-----

		2. Byte
--	--	---------

		3. Byte
--	--	---------

		LSB
--	--	-----

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu\text{s}$ / 4.8 $\mu\text{s}$	3	12 (4,4,4)

S	Undefiniert
Z	Gesetzt, wenn Bit b = 0, sonst rückgesetzt
H	Gesetzt
P/V	Undefiniert
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt von HL wird als Speicheradresse interpretiert, das Bit *b* der Zelle wird getestet; dies geschieht durch Transferieren des Komplements des zu testenden Bits in das Nullbedingungsbit (= Zero Flag Z) des Registers F.

Getestetes Bit	0	1	2	3	4	5	6	7
<i>b</i>	000	001	010	011	100	101	110	111

Beispiel: Das Registerpaar HL enthalte den Wert 4711H, Speicherstelle 4711H habe auf Bitposition 7 eine Null.  
Nach der Ausführung des Befehls

BIT 7, (HL)

ist das Nullbedingungsbit (= Z-Flag) = 1 (unabhängig von seinem bisherigen Zustand). Der Inhalt der Speicherstelle 4711H bleibt unverändert.

# BIT *b*, [IX+d]

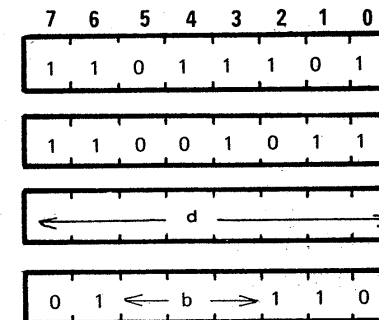
Bit in indiziert adressierter Speicherstelle testen

$Z \leftarrow \overline{(IX+d)_b}$			
Marke	Opcode	Operanden	Kommentar
Label:	BIT	<i>b</i> , (IX+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

D D	MSB
C B	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu$ s / 8 $\mu$ s	5	20 (4,4,3,5,4)

Folgende Status-Bits werden beeinflusst

S	Undefiniert
Z	Gesetzt, wenn Bit <i>b</i> = 0, sonst rückgesetzt
H	Gesetzt
P/V	Undefiniert
N	Rückgesetzt
C	Nicht beeinflusst

# BIT b, [IY+d]

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert. Das Bit b dieser Speicherzelle wird getestet, dies geschieht durch Transferieren des Komplements des zu testenden Bits in das Nullbedingungsbit (Zero-Flag Z) des Registers F.

Getestetes Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

Beispiel: Das Indexregister IX enthalte den Wert 4711H, Speicherstelle 4713H habe auf Bitposition 7 eine Null. Nach der Ausführung des Befehls

BIT 7, (IX+2)

ist das Nullbedingungsbit (Z-Flag) = 1 (unabhängig -von seinem bisherigen Zustand). Der Inhalt der Speicherstelle 4713H bleibt unverändert.

Was macht der Computer?

Befehlsformat im Assembler

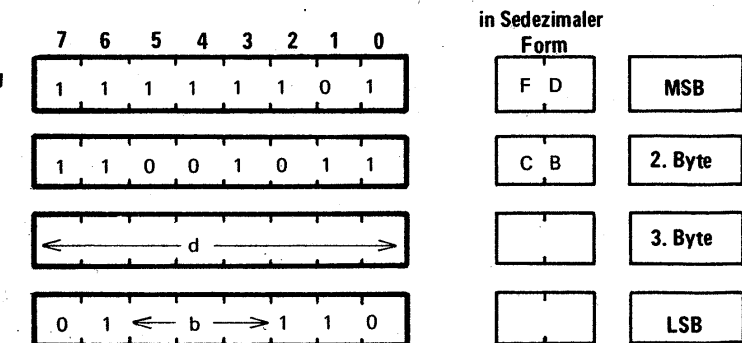
andere Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Bit in indiziert adressierter Speicherstelle testen

$Z \leftarrow \overline{(IY+d)_b}$			
Marke	Opcode	Operanden	Kommentar
Label:	BIT	b, (IY+d)	; comment



Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5 $\mu\text{s}$ / 8 $\mu\text{s}$	5	20 (4,4,3,5,4)

S	Undefiniert
Z	Gesetzt, wenn Bit b = 0, sonst rückgesetzt
H	Gesetzt
P/V	Undefiniert
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert. Das Bit b dieser Speicherzelle wird getestet; dies geschieht durch Transferieren des Komplements des zu testenden Befehls in das Nullbedingungsbit (=Zero-Flag Z) des Registers F

Getestetes Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

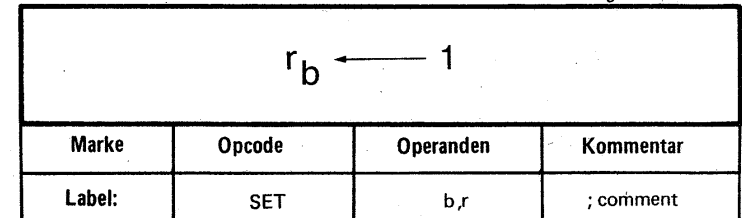
Beispiel: Das Indexregister IY enthalte den Wert 4711H, Speicherstelle 4713H habe auf Bitposition 7 eine Null.  
Nach Ausführung des Befehls

BIT 7, (IY+2)

ist das Nullbedingungsbit (Z-Flag) = 1 (unabhängig von seinem bisherigen Zustand). Der Inhalt der Speicherstelle 4713H bleibt unverändert.

# SET b, r

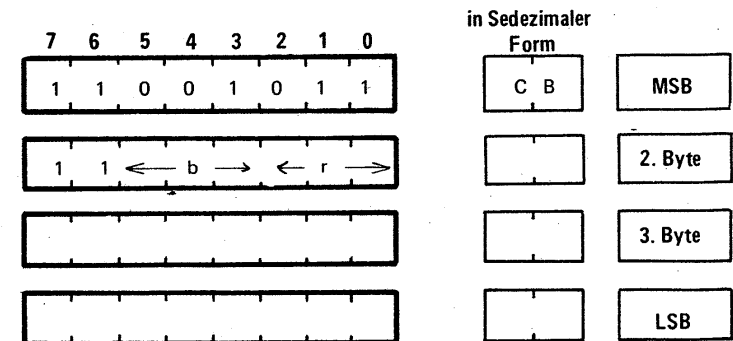
Bit in Register setzen



Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	5	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

Kommentar

Bit b des angegebenen Registers wird gesetzt.

Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

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

Beispiel: Nach dem Befehl

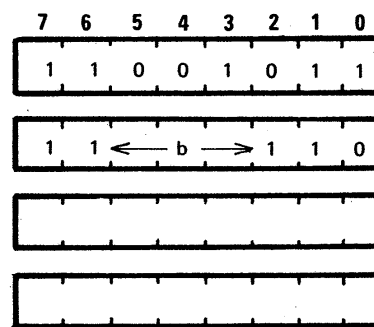
SET 4,C

ist das Bit 4 in Register C gesetzt, unabhängig von seinem bisherigen Zustand.

# SET b, [HL]

Bit in Speicherstelle setzen

(HL) <sub>b</sub> ← 1			
Marke	Opcode	Operanden	Kommentar
Label:	SET	b, (HL)	; comment



in Sedezimaler Form

C B	MSB
	2. Byte
	3. Byte
	LSB

Wicht der Operatoren?

Format Assembler

Darstellung Befehles

zur Befehls-führung

Status-Bits werden beeinflusst

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu\text{s}$ / 6 $\mu\text{s}$	4	15 (4,4,4,3)

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt von HL wird als Speicheradresse interpretiert.

Das Bit  $b$  der Zelle wird gesetzt. Die einzelnen Bit-Positionen der adressierten Speicheradresse werden im Maschinencode wie folgt verschlüsselt:

Bit	0	1	2	3	4	5	6	7
$b$	000	001	010	011	100	101	110	111

Beispiel: Der Inhalt des Registerpaares HL sei 3000H.  
Nach der Ausführung des Befehls

SET 4,HL

steht in Bitposition 4 der Speicheradresse 3000H eine 1, unabhängig vom bisherigen Wert dieses Bits.

# SET $b, [IX+d]$

Bit in indiziert adressierter Speicherstelle setzen

$(IX+d)_b \leftarrow 1$			
Marke	Opcode	Operanden	Kommentar
Label:	SET	$b, (IX+d)$	; comment

... macht der Computer?

... Format Assembler

... Darstellung Befehles

7	6	5	4	3	2	1	0	in Sedezimaler Form		
1	1	0	1	1	1	0	1	D	D	MSB
1	1	0	0	1	0	1	1	C	B	2. Byte
← d →										3. Byte
1	1	← b →		1	1	0				LSB

... zur Befehlsführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

... Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

# SET b, [IY+d]

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert.

Das Bit b der Zelle wird gesetzt. Die einzelnen Bit-Positionen der adressierten Speicheradresse werden im Maschinencode wie folgt verschlüsselt:

Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

Beispiel: Der Inhalt des Indexregisters IX sei 3000H.  
Nach der Ausführung des Befehls

SET 4, (IX+FH)

steht in Bitposition 4 der Speicheradresse 300FH eine 1, unabhängig vom bisherigen Wert dieses Bits.

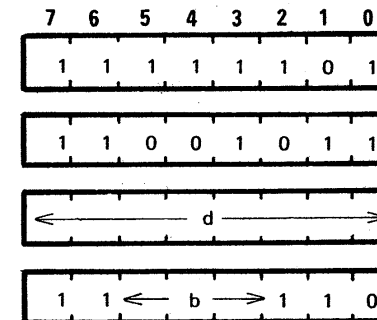
Bit in indiziert adressierter Speicherstelle setzen

$(IY+d)_b \leftarrow 1$			
Marke	Opcode	Operanden	Kommentar
Label:	SET	b, (IY+d)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls



in Sedezimaler Form

F D	MSB
C B	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement, wird als Speicheradresse interpretiert.

Das Bit b der Zelle wird gesetzt. Die einzelnen Bit-Positionen der adressierten Speicheradresse werden im Maschinencode wie folgt verschlüsselt:

Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

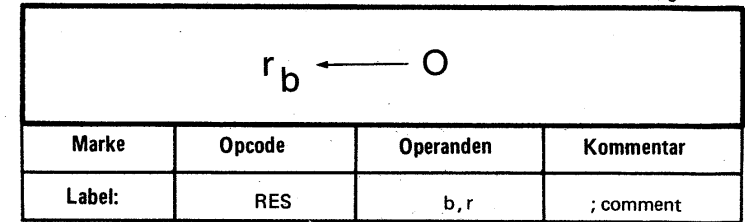
Beispiel: Der Inhalt des Indexregisters IY sei 3000H.  
Nach der Ausführung des Befehls

SET 4, (IY+FH)

steht in Bitposition 4 der Speicheradresse 300FH eine 1, unabhängig vom bisherigen Wert dieses Bits.

# RES b, r

Rücksetzen eines Bits im Register



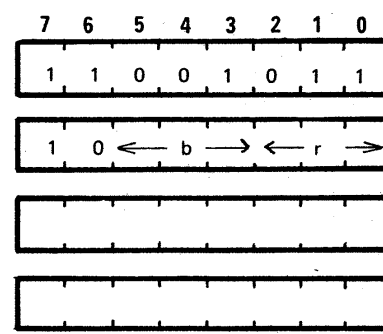
Wicht der r? r?

Format assembler

Darstellung Befehls

Wen zur Befehls-führung

folgende Status-Bits werden beeinflusst



in Sedezimaler Form

C, B	MSB
	2. Byte
	3. Byte
	LSB

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	4	8 (4,4)

S	---
Z	---
H	---
P/V	---
N	---
C	---



Kommentar

Das Bit b des Operanden wird rückgesetzt.

Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

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

Beispiel: Nach der Ausführung des Befehls

RES 5,H

ist das Bit 5 von H rückgesetzt, unabhängig von seinem bisherigen Zustand.

# RES b, [HL]

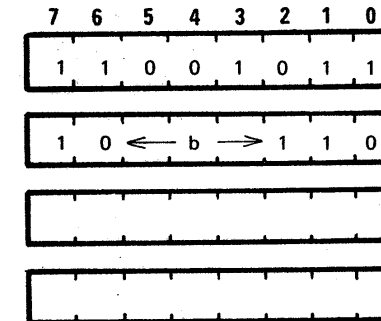
Rücksetzen eines Bits in Speicherstelle

(HL) <sub>b</sub> ← 0			
Marke	Opcode	Operanden	Kommentar
Label:	RES	b, (HL)	; comment

Was macht der Computer?

Bezeichnungsformat Assembler

Binäre-Darstellung des Befehls



in Sedezimaler Form

C, B	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.75 $\mu$ s / 6 $\mu$ s	4	15 (4,4,4,3)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

Kommentar

# RES b, [IX+d]

Der Inhalt des HL-Registerpaares wird als Speicheradresse interpretiert.

Das Bit b der Speicherzelle wird rückgesetzt.  
Die einzelnen Bitpositionen sind im Maschinencode wie folgt verschlüsselt:

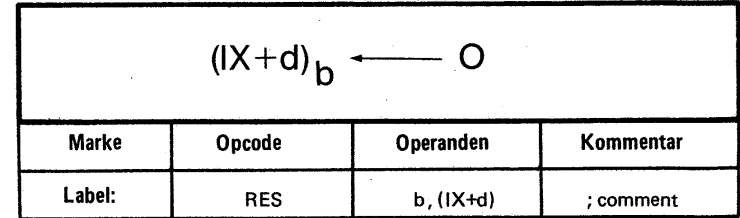
Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

Beispiel: Im Registerpaar HL stehe der Wert FFF0H.  
Nach Ausführung des Befehls

RES 0, (HL)

hat Bit 0 der Speicherstelle FFF0H den Wert 0, unabhängig von seinem bisherigen Wert.

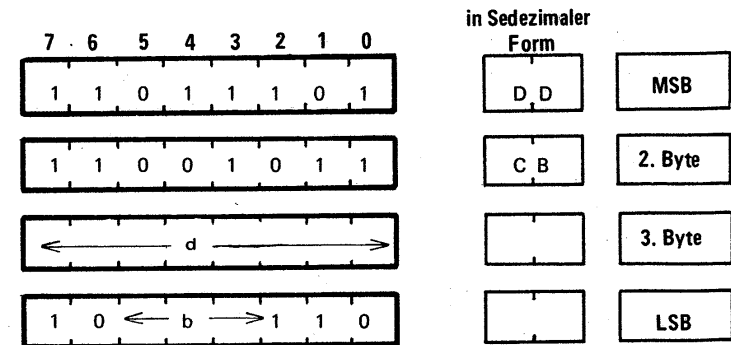
Rücksetzen eines Bits in indiziert adressierte Speicherstelle



Was macht der Computer?

Beispielformat Assembler

Binäre Darstellung des Befehls



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

Kommentar

# RES b, [IY+d]

Der Inhalt von IX, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert.

Das Bit b der Zelle wird rückgesetzt. Die einzelnen Bitpositionen sind im Maschinencode wie folgt verschlüsselt:

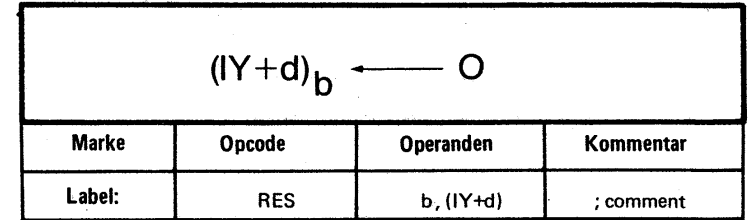
Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

Beispiel: Im Indexregister IX stehe der Wert FFF0H.  
Nach der Ausführung des Befehls

RES 0, (IX+FH)

hat Bit 0 der Speicherstelle FFFFH den Wert 0, unabhängig von seinem bisherigen Wert.

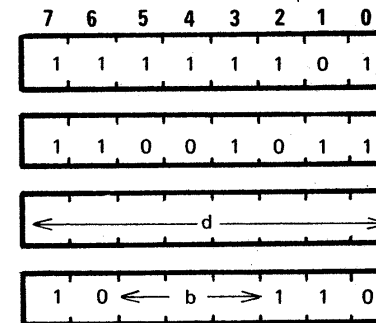
Rücksetzen eines Bits in indiziert adressierte Speicherstelle



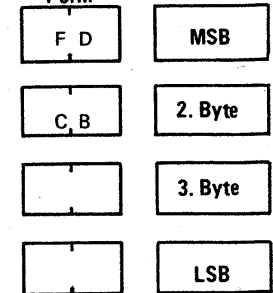
Wacht der Computer?

Assembler

Binäre Darstellung Befehls



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.75 $\mu$ s / 9.2 $\mu$ s	6	23 (4,4,3,5,4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt von IY, addiert mit d (im 2-Komplement), wird als Speicheradresse interpretiert.

Das Bit b der Zelle wird rückgesetzt. Die einzelnen Bitpositionen sind im Maschinencode wie folgt verschlüsselt:

Bit	0	1	2	3	4	5	6	7
b	000	001	010	011	100	101	110	111

Beispiel: Im Indexregister IY stehe der Wert FFF0H.  
Nach Ausführung des Befehls

RES 0, (IY+FH)

hat Bit 0 der Speicherstelle FFFFH den Wert 0, unabhängig von seinem bisherigen Wert.

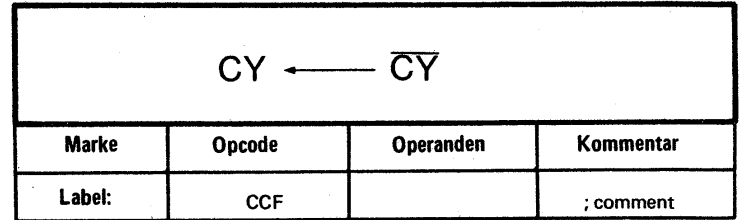
# CPU-Steuerbefehle

Complement Carry Flag, 'CCF'	3F
Set Carry Flag, 'SCF'	37
'NOP'	90
'HALT'	76
DISABLE INT '(DI)'	F3
ENABLE INT '(EI)'	FB
SET INT MODE 0 'IM0'	ED 46
SET INT MODE 1 'IM1'	ED 56
SET INT MODE 2 'IM2'	ED 5E

# CCF

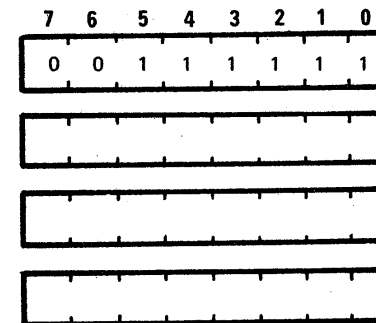
Komplementieren des Übertragsbits

Was macht der Computer?

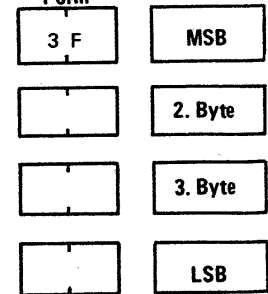


Befehlsformat  
Assembler

Binäre Darstellung  
des Befehles



in Sedezimaler  
Form



Daten zur Befehls-  
ausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s/ 1.6 $\mu$ s	1	4

Folgende Status-Bits  
werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Bisheriger Inhalt des Übertrags-(Carry) bits wird kopiert
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Gesetzt, falls CY vorher Null war, sonst rückgesetzt

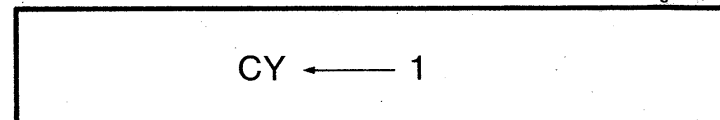
## Kommentar

Der Inhalt des Übertrags-(= "Carry")-Bedingungsbits im Register F wird komplementiert.

# SCF

Setzen des Übertragsbits

Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	SCF		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	1

in Sedezimaler Form

3 7	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Rückgesetzt
P/V	Nicht beeinflusst
N	Rückgesetzt
C	Gesetzt

## Kommentar

Der Inhalt des Übertrags-) = "Carry"-Bedingungsbits im Register F wird gesetzt.

# NOP

Leerbefehl

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	NOP		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

in Sedezimaler Form

00	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

## Kommentar

Bei Ausführung des Befehls führt die CPU keine Operation aus. Das hardwaremäßige Zeitverhalten der CPU ist jedoch wie bei den anderen Befehlen; so wird z.B. auch Refresh erzeugt.

Der Befehl kann beispielsweise als Platzhalter für evtl. später einzufügende Befehle verwendet werden.

# HALT

CPU in HALT-Zustand bringen

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Marke	Opcode	Operanden	Kommentar
Label:	HALT		; comment

7	6	5	4	3	2	1	0
0	1	1	1	0	1	1	0

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

7 6	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst



## Kommentar

Der Befehl läßt die CPU eine unbegrenzte lange Folge von Leerbefehlen (NOP) ausführen, bis ein Interrupt(=  $\overline{\text{INT}}$  oder  $\overline{\text{NMI}}$ ) oder der Rückstell (RESET) Eingang der CPU aktiv wird.

Da im HALT-Zustand Leerbefehle ausgeführt werden, ist Durchführung von Refresh gewährleistet.

# DI

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Interrupt Sperren

IFF ← O			
Marke	Opcode	Operanden	Kommentar
Label:	DI		; comment

7	6	5	4	3	2	1	0
1	1	1	1	0	0	1	1

in Sedezimaler Form

--

F 3	MSB
-----	-----

--

	2. Byte
--	---------

--

	3. Byte
--	---------

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu\text{s}$ / 1.6 $\mu\text{s}$	1	4

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

## Kommentar

Durch den Befehl wird der maskierbare Interrupt (Eingang  $\overline{INT}$ ) durch Rücksetzen des Interrupt-Freigabe-Flip-Flops IFF1 bzw. IFF2 der CPU gesperrt.

### Zur Beachtung:

Während der Ausführung des Befehls wird von der CPU kein nichtmaskierbarer Interrupt akzeptiert.

Beispiel: Nach der Ausführung des Befehls

DI

ist der maskierbare Interrupt gesperrt, bis er durch einen späteren EI-Befehl freigegeben wird. Bei gesperrtem maskierbarem Interrupt reagiert die CPU nicht auf entsprechende Interruptanforderungen.

# EI

Interrupt-Freigabe

IFF ← 1			
Marke	Opcode	Operanden	Kommentar
Label:	EI		; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	1

in Sedezimaler Form

--	--	--	--	--	--	--	--

F	B	MSB
---	---	-----

--	--	--	--	--	--	--	--

		2. Byte
--	--	---------

--	--	--	--	--	--	--	--

		3. Byte
--	--	---------

		LSB
--	--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

## Kommentar

Durch den Befehl EI wird das CPU-interne Interrupt-Freigabe-Flip-Flop (IFF1 bzw. IFF2) gesetzt.

### Zur Beachtung:

Während der Ausführung des Befehls EI und des darauffolgenden Befehls akzeptiert die CPU keine nichtmaskierbaren Interruptanforderungen.

Beispiel: Durch die Ausführung der Befehlsfolge

```

EI
RETI
    
```

ist der maskierbare Interrupt nach Ausführung der RETI-Anweisung freigegeben.

# IM O

Festlegung der Interrupt-Betriebsart

Was macht der Computer?			
Marke	Opcode	Operanden	Kommentar
Label:	IM	0	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1

0	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

E D	MSB
-----	-----

4 6	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

# Kommentar

Der Befehl bringt die CPU in die Interrupt-Betriebsart 0. In dieser Betriebsart kann die unterbrechende Peripherie hardwaremäßig beliebige Anweisungen auf den Datenbus legen, die dann von der CPU ausgeführt werden.

# IM 1

Festlegen der Interrupt-Betriebsart

...macht der Computer?

...Format Assembler

...re Darstellung des Befehles

...aten zur Befehlsführung

...igende Status-Bits werden beeinflusst

Marke	Opcode	Operanden	Kommentar
Label:	IM	1	; comment

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	0	1	0	1	1	0	5 6	2. Byte
									3. Byte
									LSB

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

# Kommentar

Der Befehl bringt die CPU in die Interrupt-Betriebsart 1. In dieser Betriebsart führt die CPU einen RESTART-Befehl zu Speicherstelle 0038H aus.

# IM 2

Festlegen der Interrupt-Betriebsart

... macht der Computer?

... Informat Assembler

... Darstellung Befehles

... zur Befehls-führung

... welche Status-Bits werden beeinflusst

Marke	Opcode	Operanden	Kommentar
Label:	IM	2	; comment

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	0	1	1	1	1	0	5 E	2. Byte
									3. Byte
									LSB

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s/ 3.2 $\mu$ s	2	8 (4,4)

S	Nicht beeinflusst
Z	Nicht beeinflusst
H	Nicht beeinflusst
P/V	Nicht beeinflusst
N	Nicht beeinflusst
C	Nicht beeinflusst

Der Befehl bringt die CPU in die Interrupt-Betriebsart 2. In dieser Betriebsart ist ein speicher-indirekt arbeitender Bedienroutinenaufruf möglich, wobei die angesprungene Bedienroutine in jedem beliebigen Bereich des 64k Byte großen Arbeitsspeichers beginnen bzw. stehen kann.

# Sprungbefehle

Bedingung

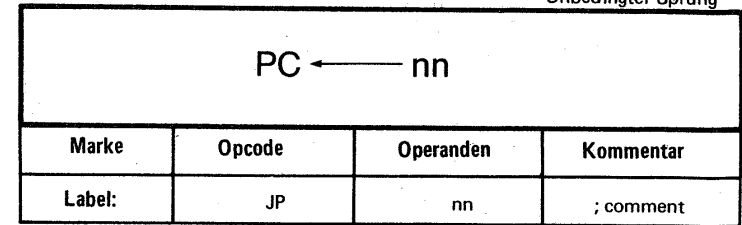
			UN-COND.	CARRY	NON CARRY	ZERO	NON ZERO	PARITY EVEN	PARITY ODD	SIGN NEG	SIGN POS	REG B≠0
'JP'	IMMED. EXT.	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
'JR'	RELATIVE	PC+e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2					
'JP'	REG. INDIR.	(HL)	E9									
'JP'		(IX)	DD E9									
'JP'		(IY)	FD E9									
	IMMED. EXT.	nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
INSTRUMENT B, NON-DANZ'	RELATIVE	PC+e										10 e-2
IN	REGISTER INDIR.	(SP) (SP+1)	09 n	D8 n	D9 n	C8 n	C9 n	E8 n	E9 n	F8 n	F9 n	
IN FROM 'SETI'	REG. INDIR.	(SP) (SP+1)	ED 4D									
IN FROM MASKABLE 'STN'	REG. INDIR.	(SP) (SP+1)	ED 45									

CALL ADDRESS		OP CODE	
0000 <sub>H</sub>	C7	'RST 0'	
0008 <sub>H</sub>	CF	'RST 8'	
0010 <sub>H</sub>	D7	'RST 16'	
0018 <sub>H</sub>	DF	'RST 24'	
0020 <sub>H</sub>	E7	'RST 32'	
0028 <sub>H</sub>	EF	'RST 40'	
0030 <sub>H</sub>	F7	'RST 48'	
0038 <sub>H</sub>	FF	'RST 56'	

Zur Beachtung: Einige Bedingungsbits haben Mehrfach-Bedeutung

# JP nn

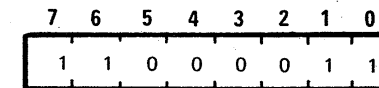
Unbedingter Sprung



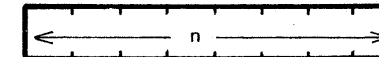
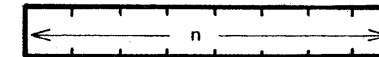
... macht der Computer?

... Assembler

... Darstellung des Befehles



in Sedezimaler Form



... zur Befehlsführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

... folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Operand nn wird in den Befehlszähler PC geladen und zeigt auf den nächsten durchzuführenden Befehl.

Beispiel: Nach Ausführung des Befehls

JP 3333H

wird das Programm ab der Adresse 3333H weitergeführt.

Achtung: Im assemblierten Befehl steht die Adresse in umgekehrter Reihenfolge, d.h. das auf den Opcode folgende Byte ist das niederwertige Adreßbyte.

Bemerkung: Das Sprungziel nn kann auch als Symbolischer Name angegeben werden.

# JP c, nn

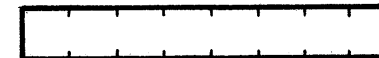
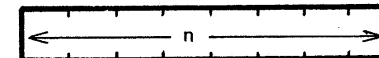
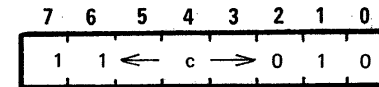
Bedingter Sprung

wenn c erfüllt ist,			
PC ← nn			
Marke	Opcode	Operanden	Kommentar
Label:	JP	C,nn	; comment

Was macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Falls die Bedingung c erfüllt ist, wird nn in den Befehlszähler geladen und das Programm läuft ab der Adresse nn weiter. Wenn c nicht erfüllt ist, wird der Befehl ignoriert.

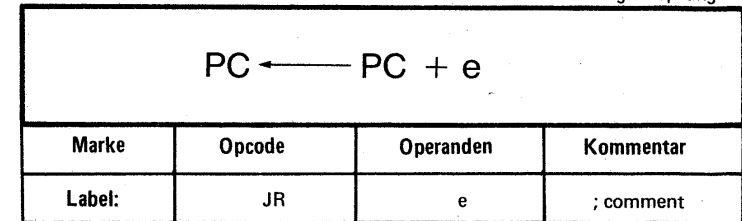
Code	000	001	010	011	100	101	110	111
Bedingung c	NZ (nicht Null)	Z (Null)	NC (kein Übertrag)	C (Übertrag)	PO (gerade Parität)	PE (ungerade Parität)	P (Vorzeichen positiv)	M (Vorzeichen negativ)
zugehöriges FLAG-Bit	Z	Z	CY	CY	P/V	P/V	S	S

**Achtung:** Im assemblierten Befehl steht die Adresse nn in umgekehrter Reihenfolge, d.h. das auf den Opcode folgende Byte ist das niederwertige Adreßbyte.

**Bemerkung:** Das Sprungziel nn kann auch als symbolischer Name angegeben werden.

# JR e

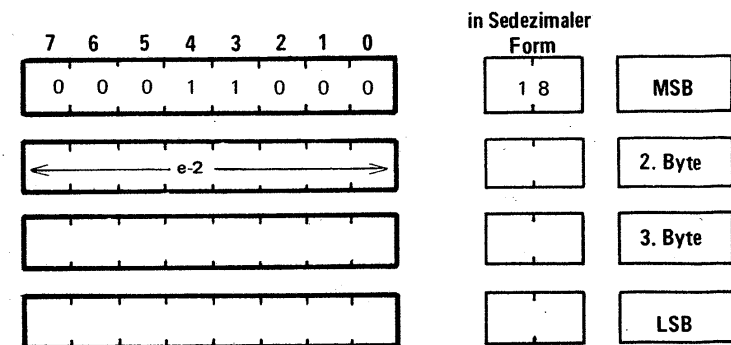
Relativer unbedingter Sprung



... macht der Computer?

... Befehlsformat Assembler

... Binäre Darstellung des Befehles



... Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu\text{s}$ / 4.8 $\mu\text{s}$	3	12 (4,3,5)

... folgende Status-Bits werden beeinflußt

S	—
Z	—
H	—
P/V	—
N	—
C	—

# JR C, e

Dieser Befehl ermöglicht den unbedingten Sprung relativ zum momentanen Befehlszählerstand. Die Sprungweite e wird zum momentanen Stand des Befehlszählers addiert (im 2-Komplement) und ermöglicht einen Sprung im Bereich zwischen -126 und 129 Bytes.

Beispiel: In Adressen 480 und 481 stehe der Befehl

JR \$ + 5

Nach Durchführung dieses Befehls wird das Programm ab der Adresse 485 weiter abgearbeitet.

Bemerkung: Statt Angabe der relativen Sprungweite e kann das Sprungziel auch symbolisch angegeben werden, z.B. JR MARKE.

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

Folgende Status-Bits werden beeinflusst

Relativer Sprung auf Carry-Bedingung

Bei C = 1

PC ← PC + e

Marke	Opcode	Operanden	Kommentar
Label:	JR	C,e	; comment

in Sedezimaler Form

7 6 5 4 3 2 1 0	0 0 1 1 1 0 0 0	3, 8	MSB
← e-2 →			2. Byte
			3. Byte
			LSB

Daten zur Befehlsausführung

Bei C = 1  
Bei C ≠ 1

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu\text{s}$ / 4.8 $\mu\text{s}$	3	12 (4,3,5)
1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Dieser Befehl erlaubt einen relativen Sprung um e Bytes vom momentanen Stand des Befehlszählers aus, wobei der Sprung nur unter der Bedingung des gesetzten CY-Bits stattfindet. Bei nicht gesetztem CY-Bit wird der Befehl ignoriert.

Die Sprungweite e wird zum momentanen Stand des Befehlszählers addiert (im 2-Komplement) und ermöglicht einen Sprung im Bereich zwischen -126 und 129 Bytes.

**Beispiel:** CY sei gesetzt und es solle um 4 Bytes von Adresse 480 rückwärts gesprungen werden.  
Der zugehörigen Befehl lautet:

JR C, \$ - 4

übersetzt -> 38 FA (-4 im 2-Komplement).  
Nach Durchführung des Befehls läuft das Programm ab Adresse 47C weiter.

**Bemerkung:** Es ist auch eine symbolische Angabe des Sprungziels möglich, z. B. JR C, MARKE.

# JR NC, e

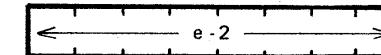
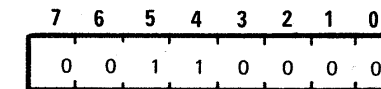
Relativer Sprung bei Nicht-Carry-Bedingung

Bei C = 0: $PC \leftarrow PC + e$			
Marke	Opcode	Operanden	Kommentar
Label:	JR	NC,e	; comment

Was macht der Computer?

Befehlsformat im Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

3 0	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Bei C = 0  
Bei C ≠ 0

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu$ s / 4.8 $\mu$ s	3	12 (4,3,5)
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Dieser Befehl erlaubt den relativen Sprung um e Bytes vom momentanen Stand des Befehlszählers, falls gleichzeitig das CY-Byte rückgesetzt war. Bei gesetztem CY-Bit wird der Befehl ignoriert.

Die Sprungweite e wird zum momentanen Stand des Befehlszählers addiert (im 2-Komplement) und ermöglicht einen Sprung im Bereich zwischen -126 und 129 Bytes.

Beispiel: CY sei rückgesetzt und es solle um 4 Bytes von Adresse 480 rückwärts gesprungen werden.  
Der zugehörige Befehl lautet:

JR NC, § - 4

Nach Überprüfung des Befehls läuft das Programm ab Adresse 47C weiter.

Bemerkung: Es ist auch eine symbolische Angabe des Sprungziels möglich, z.B. JR NC, MAR1

# JR Z, e

Relativer Sprung auf Null-Bedingung

wenn Z = 1: $PC \leftarrow PC + e$			
Marke	Opcode	Operanden	Kommentar
Label:	JR	Z,e	; comment

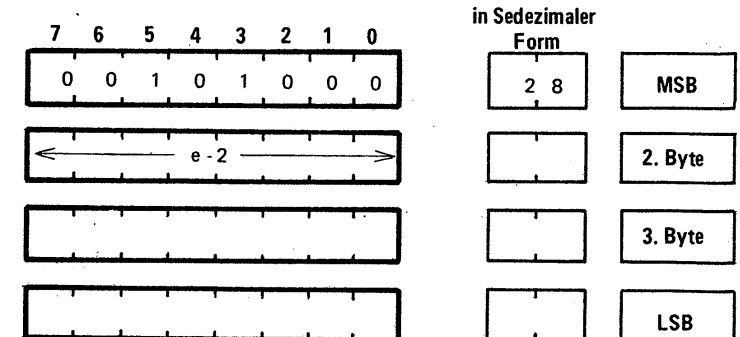
... macht der Computer?

... Format Assembler

... Darstellung des Befehles

... zur Befehlsführung

... Status-Bits werden beeinflusst



Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
Bei Z = 1 3 $\mu\text{s}$ / 4.8 $\mu\text{s}$	3	12 (4,3,5)
Bei Z $\neq$ 1 1.75 $\mu\text{s}$ / 2.8 $\mu\text{s}$	2	7 (4,3)

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Dieser Befehl erlaubt den relativen Sprung um e Bytes vom momentanen Stand des Befehlszählers, falls gleichzeitig das Z-Bit gesetzt war. Bei nicht gesetztem Z-Bit wird der Befehl ignoriert.

Die Sprungweite e wird zum momentanen Stand des Befehlszählers addiert (im 2-Komplement) und ermöglicht einen Sprung im Bereich zwischen -126 und 129 Bytes.

Beispiel: Das Nullbedingungsbit Z des Registers F sei gesetzt und es solle um 4 Bytes von Adresse 480 rückwärts gesprungen werden.

Der zugehörige Befehl lautet:

JR Z, S -4

Nach Durchführung des Befehls läuft das Programm ab Adresse 47C weiter.

Bemerkung: Es ist auch eine symbolische Angabe des Sprungziels möglich, z.B. JR Z, M5.

# JR NZ, e

Relativer Sprung auf Nicht-Null-Bedingung

wenn Z = 0:			
PC ← PC + e			
Marke	Opcode	Operanden	Kommentar
Label:	JR	NZ, e	; comment

Was macht der Computer?

Befehlsformat Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0

in Sedezimaler Form

20	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsführung

Bei Z = 0  
Bei Z ≠ 0

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu$ s / 4.8 $\mu$ s	3	12 (4,3,5)
1.75 $\mu$ s / 2.8 $\mu$ s	2	7 (4,3)

folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Dieser Befehl erlaubt den relativen Sprung um e Bytes vom momentanen Stand des Befehlszählers, falls gleichzeitig das Z-Bit rückgesetzt war.

Bei gesetztem Z-Bit wird der Befehl ignoriert.

Beispiel: Das Nullbedingungsbit Z des Registers F sei nicht gesetzt und es solle um 4 Bytes von Adresse 480 rückwärts gesprungen werden.  
Der zugehörige Befehl lautet:

JR NZ, \$ - 4

Nach Durchführung des Befehls läuft das Programm ab Adresse 47C weiter.

Bemerkung: Es ist auch eine symbolische Angabe des Sprungziels möglich, z.B. JR NZ, SPZX.

# JP [HL]

Sprung über Registerinhalt

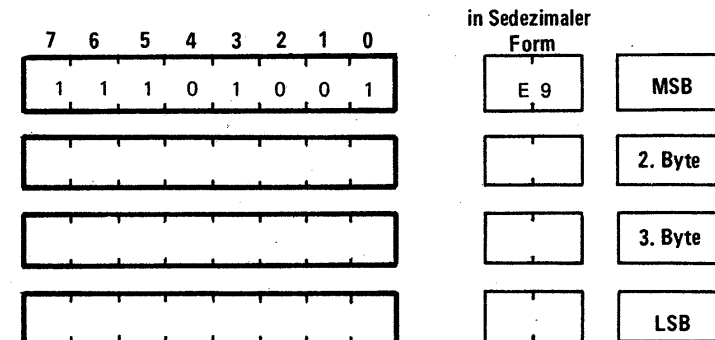
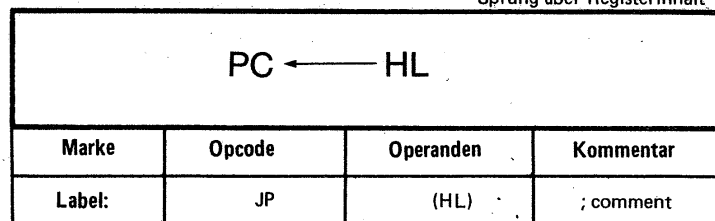
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst



Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
1 $\mu$ s / 1.6 $\mu$ s	1	4

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt des Befehlszählers wird mit dem Inhalt des HL-Registerpaars überschrieben. Dieser Befehl ermöglicht die Errechnung einer Sprungadresse durch ein Programm, nebst nachfolgender Durchführung.

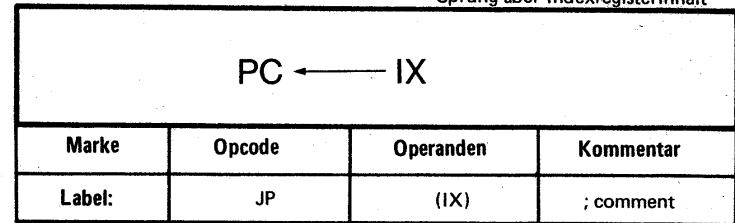
Beispiel: Der Befehlszähler enthalte 1112H, HL enthalte 3456H.  
Nach der Ausführung des Befehls

JP (HL)

enthält der Befehlszähler 3456H, das Programm wird ab dieser Stelle weitergeführt.  
Der Inhalt von HL bleibt erhalten.

# JP [IX]

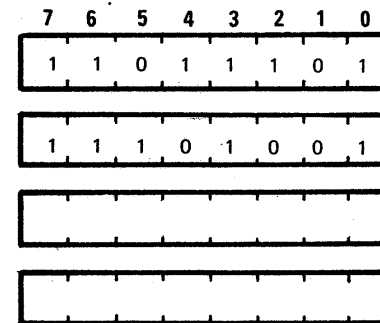
Sprung über Indexregisterinhalt



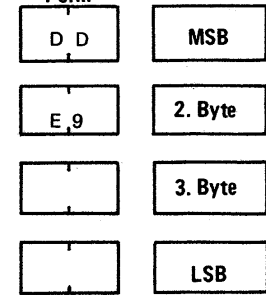
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s / 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---

## Kommentar

Der Befehlszähler wird mit dem Inhalt des Registers IX überschrieben. Das Programm wird nach dem Befehl ab der neuen Adresse weitergeführt.

Beispiel: Der Befehlszähler enthalte 1112H, IX enthalte 3456H. Nach der Ausführung des Befehls

JP (IX)

enthält der Befehlszähler 3456H, das Programm wird ab dieser Stelle weitergeführt. Der Inhalt von IX bleibt erhalten.

# JP [IY]

Sprung über Indexregisterinhalt

PC ← IY			
Marke	Opcode	Operanden	Kommentar
Label:	JP	(IY)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1

1	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

F D	MSB
-----	-----

E 9	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2 $\mu$ s/ 3.2 $\mu$ s	2	8 (4,4)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Der Befehlszähler wird mit dem Inhalt des Registers IY überschrieben. Das Programm wird nach Durchführung des Befehls ab der neuen Adresse weitergeführt.

Beispiel: Der Befehlszähler enthalte 1112H, IY enthalte 3456H. Nach der Ausführung des Befehls

JP (IY)

enthält der Befehlszähler 3456H, das Programm wird ab dieser Stelle weitergeführt. Der Inhalt von IY bleibt erhalten.

# DJNZ e

Was macht der Computer?

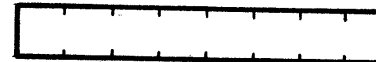
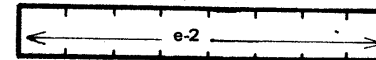
Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	DJNZ	e	; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0

in Sedezimaler Form



10	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

B ≠ 0  
B = 0

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.25 $\mu\text{s}$ / 5.2 $\mu\text{s}$	3	13 (5,3,5)
2 $\mu\text{s}$ / 3.2 $\mu\text{s}$	2	8 (5,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Dieser Befehl ist den bedingten relativen Sprungbefehlen ähnlich, mit dem Unterschied, daß der Inhalt des Registers B als Bedingung abgefragt wird.

Bei jedem Erreichen des Befehls wird der Inhalt von B um 1 erniedrigt, falls  $B \neq 0$ , führt das Programm den Sprung durch, falls  $B = 0$ , wird der Befehl ignoriert.

Beispiel: Ausgabe von 16 Leerzeichen (ASCII ' ') auf die I/-Adresse 12H

```

START: LD B,16      ; Laden des Schleifenzählers
        LD A,20H    ; Laden des Zeichens

LOOP:  OUT 12H,A    ; Ausgabe
        DJNZ LOOP  ; Rücksprung

END:   HALT        ; Ende der Ausgabe
    
```

Bemerkung: Das Sprungziel kann sowohl über Distanzangabe, als auch durch eine symbolische Marke vorgenommen werden.

# CALL nn

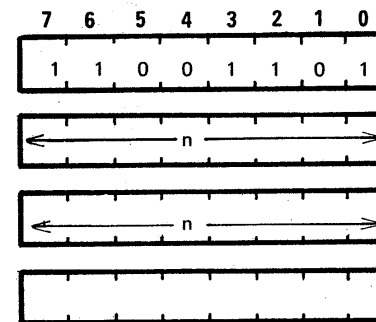
Unterprogrammaufruf

$(SP-1) \leftarrow PC_H, (SP-2) \leftarrow PC_L,$ $PC \leftarrow nn,$			
Marke	Opcode	Operanden	Kommentar
Label:	CALL	nn	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

C D	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.2 $\mu$ s / 6.8 $\mu$ s	5	17 (4,3,4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Nach Erreichen dieses Befehls wird zuerst der Befehlszählerstand in den Stack gerettet. Anschließend überschreibt nn den Befehlszählerinhalt, wodurch das Programm nun bei der Adresse nn weiterläuft.

Die bei nn beginnende Subroutine muß durch RET abgeschlossen sein, dieser Befehl restauriert den Befehlszähler aus dem Stack wieder auf die nächste Adresse des Hauptprogrammes.

Wegen der Länge des Befehls von 3 Bytes wird der Befehlszähler innerhalb des Befehls noch 3x inkrementiert, bevor er in den Stack gelangt. Die so gerettete Adresse ist die Adresse des nächsten Befehls des Hauptprogramms.

Beispiel: Beginnt der CALL-Befehl bei Adresse 2042H des Hauptprogramms, wird die Adresse 2045H im Stack gespeichert.

Enthält der Stack-Pointer von der Befehlsdurchführung 1234H, ist PC<sub>H</sub> auf 1233H, PC<sub>L</sub> auf 1232H gespeichert.

Der Stack-Pointer steht anschließend auf 1232H.

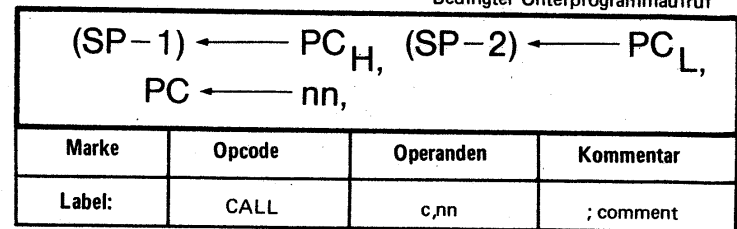
Bemerkung: Statt nn kann auch eine symbolische Marke vorgegeben werden z.B. CALL UNTPROG

Zur besonderen Beachtung:

Das auf den Operationscode folgende Byte ist das niederwertige, das nächste Byte das höherwertige Byte der Aufrufadresse.

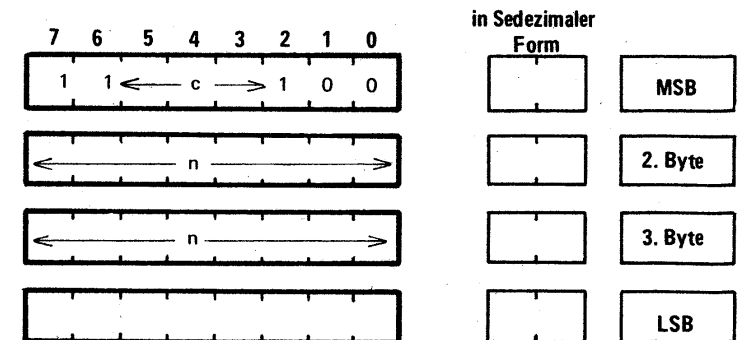
# CALL c, nn

Bedingter Unterprogrammaufruf



Befehlsformat  
in Assembler

Binäre Darstellung  
des Befehles



Daten zur Befehls-  
durchführung

c wahr  
c falsch

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4.25 $\mu\text{s}$ / 6.8 $\mu\text{s}$	5	17 (4,3,4,3,3)
2.5 $\mu\text{s}$ / 4 $\mu\text{s}$	3	10 (4,3,3)

Folgende Status-Bits  
werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Bei erfüllter Bedingung c wird der Befehlszähler mit der Adresse des nachfolgenden Befehls des Hauptprogramms in den Stack gerettet. Anschließend überschreibt nn den Befehlszähler, wobei nn die erste Adresse der gerufenen Subroutine ist.

Bei nicht erfüllter Bedingung c wird der Befehl ignoriert.

c	000	001	010	011	100	101	110	111
Symbol für Bedingung	NZ	Z	NC	C	PO	PE	P	M
zugehöriges Bedingungsbit	Z	Z	CY	CY	P/V	P/V	S	S

Beispiel: CY sei rückgesetzt, SP enthalte 3002H, PC enthalte 1478H und an dieser Adresse beginne:

CALL NC,1223H

Anschließend enthält SP 3000H  
 Speicheradresse 3001 14H  
 3000 7BH  
 und der Befehlszähler 1223H

Bemerkung: Statt nn kann auch eine symbolische Marke angegeben werden, z.B.

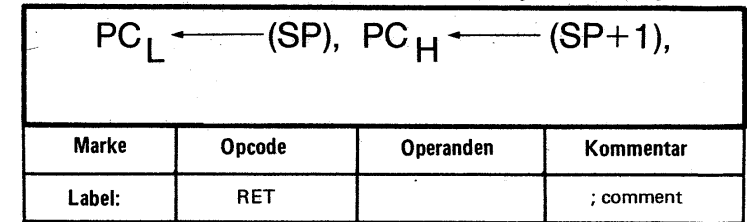
CALL PE,UP2

Zur besonderen Beachtung:

Das auf den Operationscode folgende Byte ist das niederwertige, das nächstfolgende Byte das höherwertige Byte der Aufrufadresse.

# RET

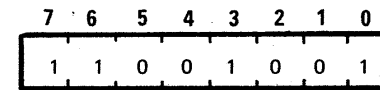
Rücksprung vom Unterprogramm



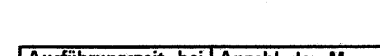
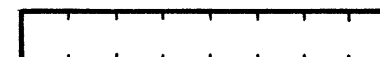
Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sederzimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.5 $\mu$ s / 4 $\mu$ s	3	10 (4,3,3)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Return-Befehl beendet ein Unterprogramm. Mit diesem Befehl wird der in den Stack gerettete Befehlszählerinhalt (durch den CALL-Befehl) wieder in den Befehlszähler zurückgeschrieben.

Damit zeigt der Befehlszähler auf den ersten Befehl des Hauptprogrammes nach dem CALL-Befehl, mit dem die nun beendete Subroutine aufgerufen wurde.

Beispiel: SP enthalte 4343H, die Speicheradresse 4343H enthalte 24, die Adresse 4344H enthalte 24, der Befehlszähler PC 2211.

Nach der Ausführung des Befehls

RET

enthält SP 4345H  
PC 2424H

Der nächste durchgeführte Befehl steht daher auf Adresse 2424.

# RET c

Bedingter Rücksprung vom Unterprogramm

falls Bedingung c erfüllt ist, gilt:

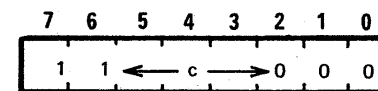
$PC_L \leftarrow (SP), PC_H \leftarrow (SP+1),$

Was macht der Computer?

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	RET	c	; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

c gilt  
c falsch

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu\text{s}$ / 4.4 $\mu\text{s}$	3	11 (5,3,3)
1.25 $\mu\text{s}$ / 2 $\mu\text{s}$	1	5

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Mit RET c ist ein bedingter Rücksprung aus einer Subroutine möglich. Gilt c, erfolgt der normale Rücksprung (siehe RET-Befehl), ist c nicht erfüllt, wird der Befehl ignoriert.

c	000	001	010	011	100	101	110	111
Symbol für Bedingung	NZ	Z	NC	C	PO	PE	P	M
zugehöriges Bedingungsbit	Z	Z	CY	CY	P/V	P/V	S	S

Beispiel: Das S-Bit sei gesetzt, SP enthalte 5545H, die Speicheradressen 5545H und 5546H enthalten 12H und 14H. Nach der Ausführung des Befehls

RET M

enthält der Befehlszähler PC 1412H und der Stackpointer SP 5547H.

# RETI

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Rückkehr von INT -Bedienprogramm			
Marke	Opcode	Operanden	Kommentar
Label:	RETI		; comment

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	0	0	1	1	0	1	4 D	2. Byte
									3. Byte
									LSB

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu\text{s}$ / 5.6 $\mu\text{s}$	4	14 (4,4,3,3)

S	---
Z	---
H	---
P/V	---
N	---
C	---

Mit RETI erfolgt der Rücksprung aus einer INTERRUPT-Subroutine.

Zwei Aufgaben werden durch RETI erfüllt:

1. Der Programmzähler wird analog zu RET restauriert.
2. Dem Peripheriebaustein, der den INT anmeldete, wird das Ende seiner zugehörigen Routine mitgeteilt. Der Baustein gibt daraufhin die vom ihm blockierte DAISY-CHAIN wieder frei und ermöglicht damit die Abarbeitung niederwertiger INTs.  
Darüberhinaus wird der Inhalt des IFF2- in das IFF1-Flip Flop kopiert.

Zur Beachtung: Durch die RETI-Anweisung wird der maskierbare Interrupt nicht freigegeben; daher sollte grundsätzlich vor jedem RETI-Befehl ein EI-Befehl stehen, der die Annahme später folgender Interruptanforderungen ermöglicht.

# RETN

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Rückkehr von NMI-Bedienprogramm			
Marke	Opcode	Operanden	Kommentar
Label:	RETN		; comment

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	0	0	0	1	0	1	4 5	2. Byte
									3. Byte
									LSB

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3.5 $\mu$ s / 5.6 $\mu$ s	4	14 (4,4,3,3)

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

RETN wird zur Rückkehr aus einer Subroutine verwendet, die durch einen NMI aufgerufen wurde (NMI = Nicht maskierbarer Interrupt).

RETN ist in seiner Wirkung vollkommen identisch zum Befehl RET, zusätzlich kopiert das IFF2-Flip-Flop sich in das IFF1-Flip-Flop, so daß die Bearbeitung maskierbarer Interruptanforderungen unmittelbar nach Ausführung des RETN-Befehls freigegeben ist, falls sie bereits vor Auftreten der NMI-Anforderung freigegeben war.

# RST p

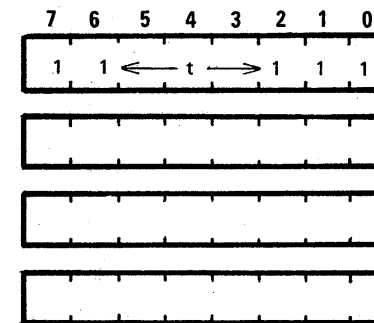
Ansprung von Service-Routinen

$(SP-1) \leftarrow PC_H, (SP-2) \leftarrow PC_L,$ $PC_H \leftarrow O, PC_L \leftarrow p,$			
Marke	Opcode	Operanden	Kommentar
Label:	RST	p	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu$ s / 4.4 $\mu$ s	3	11 (5,3,3)

Folgende Status-Bits werden beeinflußt

S	—
Z	—
H	—
P/V	—
N	—
C	—



## Kommentar

Die 8 verschiedenen RST p Befehle entsprechen Sprüngen zu Subroutinen, die an bestimmten Adressen der ersten 1k-Speicheradressen liegen. Die Adressen werden durch einen 3Bit-Code innerhalb des Befehlsbytes ausgewählt. Ansonsten entspricht RST p in seiner Wirkung voll einem normalen CALL-Befehl.

symbolische Darstellung	0	1	2	3	4	5	6	7
Adresse p	00H	08H	10H	18H	20H	28H	30H	38H
t	000	001	010	011	100	101	110	111

Bemerkung: t ist der zur Adresse p gehörige Binärcode.

# Ein/Ausgabe-Befehle

		Portadresse		
			IMMED.	REG. INDIR.
			(n)	(C)
Ziel:	REG ADDRESSING	A	DB	ED 78
		B		ED 40
		C		ED 48
		D		ED 50
		E		ED 58
		H		ED 60
		L		ED 68
Eingabe-Befehl	REG, INDIR	(HL)		ED A2
				ED B2
				ED AA
				ED BA

} Block-Eingabe-Befehle

Quelle

# IN A, [n]

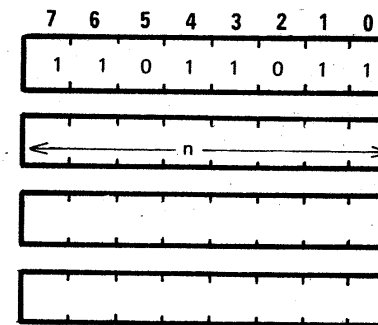
Eingabebefehl

A ← (n)			
Marke	Opcode	Operanden	Kommentar
Label:	IN	A,(n)	; comment

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form

DB	MSB
	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu\text{s}$ / 4.4 $\mu\text{s}$	3	11 (4,3,4)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

Ausgabebefehle

Portadresse

	REGISTER							REG. IND.			
	A	B	C	D	E	H	L				
'OUT'									REG. IND.	(n)	D3
'OUTI' - OUTPUT Inc HL, Dec b									REG. IND.	(C)	ED 79
'OTIR' - OUTPUT, Inc HL, Dec B, REPEAT IF B≠0									REG. IND.	(C)	ED 41
'OUTD' - OUTPUT Dec HL & B									REG. IND.	(C)	ED 49
'OTDR' - OUTPUT, Dec HL & B, REPEAT IF B≠0									REG. IND.	(C)	ED 51
									REG. IND.	(C)	ED 59
									REG. IND.	(C)	ED 61
									REG. IND.	(C)	ED 69
									REG. IND.	(C)	ED 8B

Block-Ausgabe-Befehle

## Kommentar

Der Operand n belegt die unter Hälfte des Adreßbusses und wählt eine I/O-Schnittstelle mit der Adresse n aus, der Inhalt von A belegt gleichzeitig die obere Adreßbushälfte (A8 bis A15).

Anschließend wird ein Byte der ausgewählten I/O-Schnittstelle auf den Datenbus übernommen und in den Akkumulator transferiert.

Beispiel: A enthalte 23H, die Schnittstelle mit Adresse 11H enthalte 67H.

Nach der Ausführung des Befehls

IN A, (11H)

enthält der Akkumulator 67H.

Bemerkung: Anstelle der hexadezimalen Schnittstellenadresse kann auch ein symbolischer Kanal-Name stehen.  
Beispiel: IN A, (TPPRT).

# IN r, [C]

Was macht der Computer?

Befehlsformat in Assembler

Binäre Darstellung des Befehls

Daten zur Befehlsausführung

Folgende Status-Bits werden beeinflusst

Eingabebefehl

r ← (C)			
Marke	Opcode	Operanden	Kommentar
Label:	IN	r,(C)	; comment

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	E D	MSB
0	1	← r →		0	0	0			2. Byte
									3. Byte
									LSB

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu$ s / 48 $\mu$ s	3	12 (4,4,4)

Bit	Bedeutung
S	Gesetzt, wenn Eingabe negativ, sonst rückgesetzt
Z	Gesetzt, wenn Eingabe Null, sonst rückgesetzt
H	Rückgesetzt
P/V	Gesetzt bei gerader Parität der Eingabe
N	Rückgesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des Registers C erscheint auf der unteren Adreßbushälfte (A<sub>0</sub> bis A<sub>7</sub>), Register B auf der oberen (A<sub>8</sub> bis A<sub>15</sub>).

Die durch den Inhalt des C-Registers adressierte Schnittstelle liefert das in ihr enthaltene Byte über den Datenbus in das gewünschte Register r.

REG	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

Beispiel: Register C enthalte 07H, B enthalte 10H und das Byte 77H sei in der Schnittstelle 07 verfügbar.  
Nach der Ausführung des Befehls

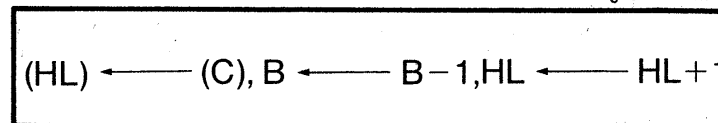
IN D, (C)

enthält das Register D 77H.

# INI

Block-Eingabe-Befehl

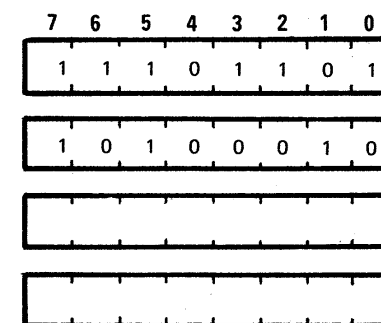
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	INI		; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

ED	MSB
A2	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflußt

S	Unbestimmt
Z	Gesetzt, wenn B-1 = 0, sonst rückgesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflußt

## Kommentar

Die Schnittstelle, deren Adresse in C enthalten ist, wird abgefragt. Ihr Inhalt gelangt über den Datenbus in jene Speicherzelle, deren Adresse im HL-Registerpaar enthalten ist. Anschließend wird der Inhalt von HL inkrementiert, der des Registers B dekrementiert.

Beispiel: C enthalte 07H, B sei 10H, HL enthalte 1000H, die Schnittstelle 07 liefere das Byte 7BH.  
Nach Ausführung des Befehls

INI

ergeben sich folgende Verhältnisse:

Speicherstelle	Reg. H	Reg. L	Reg. B
1000H			
B7H	10H	01H	0FH

# INIR

Block-Eingabe-Befehl

Was macht der Computer?

solange B ≠ 0 ist:			
$(HL), \leftarrow (C) \quad B \leftarrow B - 1,$ $HL \leftarrow HL + 1$			
Marke	Opcode	Operanden	Kommentar
Label:	INIR		; comment

Befehlsformat in Assembler

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0	in Sedezimaler Form	
1	1	1	0	1	1	0	1	ED	MSB
1	0	1	1	0	0	1	0	B 2	2. Byte
									3. Byte
									LSB

Daten zur Befehlsausführung

bei B ≠ 0  
bei B = 0

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.25 $\mu\text{s}$ / 8.4 $\mu\text{s}$	5	21 (4,5,3,4,5)
4 $\mu\text{s}$ / 6.4 $\mu\text{s}$	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

**Kommentar**

INIR ist eine fortlaufende Aktion des INI-Befehls. Die gesamten Operationen des INI-Befehls werden so lange durchgeführt, bis der Inhalt des B-Registers = 0. Ist B ≠ 0, wird PC um 2 erniedrigt und der Befehl INI erneut durchgeführt. Interruptanforderungen werden nach jedem Datentransfer angenommen und außerdem zwei Refresh-Zyklen ausgesandt.

Mit diesem Befehl kann eine fortlaufende Dateneingabe über eine Schnittstelle, deren Adresse in C steht, durchgeführt werden. Die Anzahl der übernommenen Bytes wird mit B festgelegt, die Startadresse des Datenblocks ist durch den Inhalt von HL gegeben.

Nach dem Einlesen eines jeden Datenbytes kann eine Interruptanforderung akzeptiert werden. Außerdem werden 2 Refreshzyklen durchgeführt.

Die nachfolgende Tabelle zeigt die Verhältnisse in einer typischen Situation vor und nach Ausführung des Befehls

**INIR**

Beispiel: Am Eingabeport mit der Adresse 07H sollen nacheinander die Datenwörter 51H, A9H und 03H anstehen und folgende Verhältnisse in Speicher und Registerstellen gegeben sein:

Reg. C	Reg. B	Reg. H	Reg. L	Speicherstelle 1000H	Speicherstelle 1001H	Speicherstelle 1001H
07H	03H	10H	00H	beliebig	beliebig	beliebig

Nach Ausführung des Befehls

**INIR**

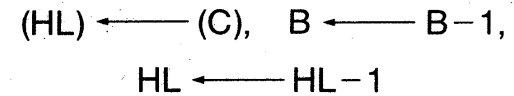
ergeben sich dann folgende Verhältnisse:

Reg. C	Reg. B	Reg. H	Reg. L	Speicherstelle 1000H	Speicherstelle 1001H	Speicherstelle 1002H
07H	0H	10H	03H	51H	A9H	03H

# IND

Block-Eingabe-Befehle

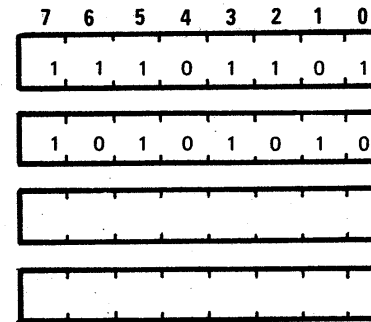
Was macht der Computer?



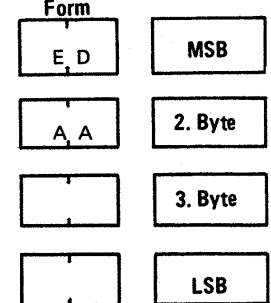
Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	IND		; comment

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s/ 6.4 $\mu$ s	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt, wenn B-1 = 0, sonst rückgesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

IND entspricht in seiner Funktion dem Befehl INI, nur wird anschließend an die Dateneingabe das HL-Registerpaar dekrementiert. Die Dateneingabe erfolgt über die Schnittstelle mit der Adresse in C, das eingegebene Byte wird im Speicher mit der Adresse (HL) abgelegt.

Beispiel: C enthalte 55H, HL enthalte 1234H, B sei 5 und in Schnittstelle 55 sei das Byte 43H verfügbar.  
Nach Ausführung des Befehls

IND

ergeben sich folgende Verhältnisse:

Speicherstelle	Reg. H	Reg. L	Reg. B
1234H	12H	33H	04H

# INDR

Block-Eingabe-Befehle

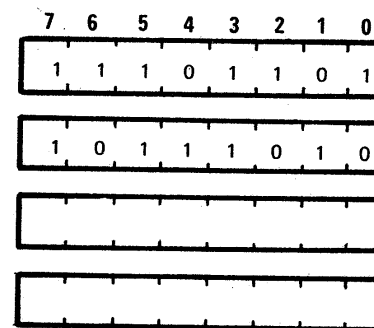
Was macht der Computer?

(HL) ← (C), B ← B - 1,  
HL ← HL - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	INDR		; comment

Binäre Darstellung des Befehles



in Sedezimaler Form

E D	MSB
B A	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

bei B ≠ 0  
bei B = 0

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.25 $\mu\text{s}$ / 8.4 $\mu\text{s}$	5	21 (4,5,3,4,5)
4	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

**Kommentar**

INDR entspricht einer mehrfachen Wiederholung des IND-Befehls. Nach jedem Datentransfer aus der Schnittstelle, deren Adresse in C enthalten ist, in eine Speicherzelle mit der Adresse (HL), wird das Register B dekrementiert, ebenso das HL-Registerpaar.

Sofern B ≠ 0, erniedrigt sich der Inhalt des Befehlszählers um 2 und der IND-Befehl wird wiederholt.

Wenn B = 0, ist der Befehl abgearbeitet.

Nach dem Einlesen eines jeden Datenbytes kann eine Interruptanforderung akzeptiert werden, außerdem werden zwei Refres-Zyklen durchgeführt.

Mit diesem Befehl wird eine fortlaufende Dateineingabe über ein Port durchgeführt, dessen Adresse im Register C steht.

Die Anzahl der übernommenen Bytes ist durch den Inhalt von Register B gegeben und die Anfangsadresse des Datenblocks durch den Inhalt des Registerpaars HL.

Beispiel: Am Eingabeport mit der Adresse 07H sollen nacheinander die Datenwörter 51H, A9H und 03H anstehen und folgende Verhältnisse in Speicher und Registerstellen gegeben sein:

Reg. C	Reg. B	Reg. H	Reg. L	Speicherstelle 1000H	Speicherstelle 1001H	Speicherstelle 1002H
07H	03H	10H	02H	beliebig	beliebig	beliebig

Nach Ausführung des Befehls

INDR

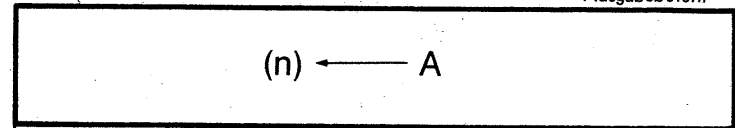
ergeben sich dann folgende Verhältnisse:

Reg. C	Reg. B	Reg. H	Reg. L	Speicherstelle 1000H	Speicherstelle 1001H	Speicherstelle 1002H
07H	0H	0FH	FFH	03H	A9H	51H

# OUT [n], A

Ausgabebefehl

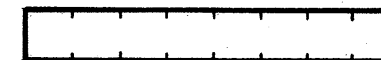
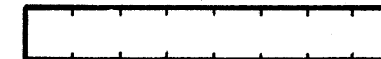
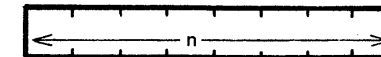
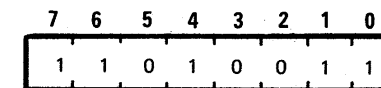
Was macht der Computer?



Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	OUT	(n),A	; comment

Binäre Darstellung des Befehls



in Sedezimaler Form

D 3	MSB
-----	-----

	2. Byte
--	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
2.75 $\mu$ s / 4.4 $\mu$ s	3	11 (4,3,4)

Folgende Status-Bits werden beeinflusst

S	---
Z	---
H	---
P/V	---
N	---
C	---



## Kommentar

Der Operand n wird auf die untere Adreßbushälfte (A<sub>0</sub> bis A<sub>7</sub>) plaziert, der Inhalt von A auf die obere A<sub>8</sub> bis A<sub>15</sub>).

Anschließend gelangt der Inhalt von A auf den Datenbus und wird über die Schnittstelle mit der Adresse n ausgegeben.

Beispiel: A enthalte 44H.  
Nach Ausführung des Befehls

OUT 12H,A

hat der Ausgabepuffer des Ports 12H den Inhalt 44H.

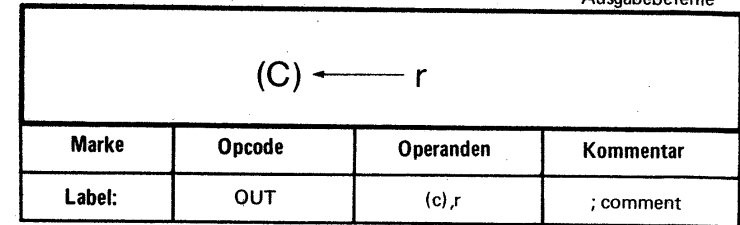
Bemerkung: Statt des Hexadezimalwertes n kann auch eine symbolische Kanaladresse angegeben werden, z.B.

OUT (OTDRVR), A

# OUT [C], r

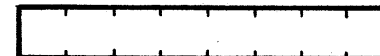
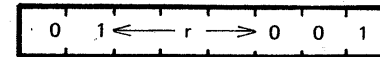
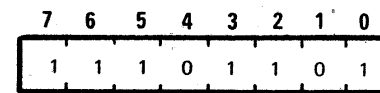
Ausgabebefehle

Was macht der Computer?



Befehlsformat in Assembler

Binäre Darstellung des Befehles



in Sedezimaler Form



Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
3 $\mu$ s / 4.8 $\mu$ s	3	12 (4,4,4)

Folgende Status-Bits werden beeinflusst

S	—
Z	—
H	—
P/V	—
N	—
C	—

## Kommentar

Der Inhalt des Registers C liefert eine Schnittstellenadresse, über die der Inhalt des Registers r ausgegeben wird. Der Adreßbus wird in der unteren Hälfte mit dem Inhalt von C, in der oberen mit dem Inhalt von B belegt.

Reg.	B	C	D	E	H	L	A
r	000	001	010	011	100	101	111

Beispiel: Das Register C enthalte 22H, Register D 4AH.  
Nach der Ausführung des Befehls

OUT (C), D

hat der Ausgabepuffer des Ports 22H den Inhalt 4AH.

# OUTI

Block-Ausgabebefehl

Was macht der Computer?

(C) ← (HL), B ← B-1,  
HL ← HL+1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	OUTI	\	; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

in Sedezimaler Form

E D	MSB
-----	-----

A 3	2. Byte
-----	---------

	3. Byte
--	---------

	LSB
--	-----

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt, wenn B-1 = 0, sonst rückgesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des HL-Registerpaares wählt eine Speicheradresse, deren Inhalt über die Schnittstelle, deren Adresse in C enthalten ist, ausgegeben.

Nach Übernahme des Inhalts der Zelle in die CPU wird der Bytezähler B dekrementiert. Anschließend gelangt der Inhalt von C auf die untere Adreßbushälfte, der von B auf die obere. Der zwischengespeicherte Inhalt der Speicherzelle wird nun auf den Datenbus gebracht und über die ausgewählte Schnittstelle ausgegeben.

Nach der Ausgabe wird das Registerpaar HL inkrementiert.

Beispiel: HL enthalte 1000H, B sei 10H, C sei 7 und der Speicher 1000H sei 77.

Nach der Ausführung des Befehls

OUTI

enthält HL 1001H und B enthält 0FH und 77H wurde über die Schnittstelle 07H ausgegeben.

# OTIR

Block-Ausgabebefehl

Was macht der Computer?

(C) ← (HL), B ← B - 1,  
HL ← HL + 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	OTIR		; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	1	0	0	1	1

in Sedezimaler Form

E D	MSB
B, 3	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

bei B ≠ 0  
bei B = 0

Ausführungszeit bei $\phi = 4/2.5 \text{ MHz}$	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.25 $\mu\text{s}$ / 8.4 $\mu\text{s}$	5	21 (4,5,3,4,5)
4 $\mu\text{s}$ / 6.4 $\mu\text{s}$	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflußt

S	Unbestimmt
Z	Gesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflußt

## Kommentar

Der Befehl OTIR entspricht einem mehrfach wiederholten OUTI-Befehl. Nach jeder Transferierung eines Bytes wird der Inhalt des dekrementierten B-Registers geprüft. Ist  $B \neq 0$ , wird PC um 2 dekrementiert und ein neuerlicher OUTI-Befehl durchlaufen, falls  $B = 0$ , ist der OTIR-Befehl zu Ende.

Nach der Ausgabe eines jeden Datenbytes werden Interrupt-Anforderungen akzeptiert, außerdem werden 2 Refres-Zyklen durchgeführt.

Beispiel: Vor bzw. nach Ausführung des Befehls

### OTIR

seien folgende Verhältnisse gegeben:

	Reg.B	Reg.C	Reg.H	Reg.L	Speicher- stelle 1000H	Speicher- stelle 1001H	Speicher- stelle 1002H
vor Aus- führung	03H	07H	10H	00H	51H	A9H	03H
nach Aus- führung	00H	07H	10H	03H	51H	A9H	03H

An dem Ausgabeport, das durch die Hexadezimalzahl 07H adressiert ist, sind nacheinander die Wert 51H, A9H 03H erschienen.

# OUTD

Block-Ausgabebefehl

Was macht der Computer?

(C) ← (HL), B ← B - 1,  
HL ← HL - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	OUTD		; comment

Binäre Darstellung des Befehles

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	0	1	0	1	1

in Sedezimaler Form

E D	MSB
A B	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
4 $\mu$ s / 6.4 $\mu$ s	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt, wenn $B = 0$ , sonst rückgesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Der Inhalt des HL-Registerpaares wird auf den Adreßbus gelegt.

Der Inhalt der ausgewählten Speicherzelle wird in der CPU zwischengespeichert, anschließend belegt der Inhalt von C die untere Adreßbushälfte, der Inhalt des in der Zwischenzeit dekrementierten B-Registers die obere. Über die durch C ausgewählte Schnittstelle wird nun der Inhalt der Speicherstelle ausgegeben. Danach wird HL dekrementiert.

Beispiel: C enthalte 66H, B 44H. Die Speicherstelle 3344H enthält 38H, im HL-Registerpaar stehe 3344H.  
Nach der Ausführung des Befehls

OUTD

enthält B 43H, HL 3343H und über die Schnittstelle 66H wurde das Byte 38H ausgegeben.

# OTDR

Block-Ausgabebefehl

Was macht der Computer?

solange B ≠ 0 ist: (C) ← (HL),  
B ← B - 1, HL ← HL - 1

Befehlsformat in Assembler

Marke	Opcode	Operanden	Kommentar
Label:	OTDR		; comment

Binäre Darstellung des Befehls

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
1	0	1	1	1	0	1	1

in Sedezimaler Form

E D	MSB
B, B	2. Byte
	3. Byte
	LSB

Daten zur Befehlsausführung

bei B ≠ 0  
bei B = 0

Ausführungszeit bei $\phi = 4/2.5$ MHz	Anzahl der Maschinen-(M-)Zyklen	Anzahl der Takt-(=T-)Zyklen und ihre Aufteilung auf die Maschinenzyklen
5.25 $\mu$ s / 8.4 $\mu$ s	5	21 (4,5,3,4,5)
4	4	16 (4,5,3,4)

Folgende Status-Bits werden beeinflusst

S	Unbestimmt
Z	Gesetzt
H	Unbestimmt
P/V	Unbestimmt
N	Gesetzt
C	Nicht beeinflusst

## Kommentar

Der Befehl OTDR entspricht einem mehrfach ausgeführten OUTD-Befehl.

Nach jedem Datentransfer wird der Inhalt des B-Registers geprüft, wenn  $B \neq 0$ , wird der Befehlszähler um 2 dekrementiert und ein neuerlicher OUTD-Befehl durchgeführt. Falls  $B = 0$ , ist der OTDR-Befehl beendet.

Nach jedem Datentransfer werden Interrupt-Anforderungen akzeptiert, außerdem werden zwei Refresh-Zyklen durchgeführt.

Beispiel: Vor bzw. nach Ausführung des Befehls

### OTDR

seien folgende Verhältnisse gegeben:

	Reg.B	Reg.C	Reg.H	Reg.L	Speicher- stelle 0FFEh	Speicher- stelle 0FFFh	Speicher- stelle 1000h
vor Aus- führung	03H	07H	10H	00H	51H	A9H	03H
nach Aus- führung	00H	07H	0FH	FDH	51H	A9H	03H

An dem Ausgabeport, das durch die Hexadezimalzahl 07H adressiert ist, sind nacheinander die Werte 03H, A9H und 51H erschienen.

## Literatur- und Quellenverzeichnis

- Zilog Data Book von Zilog GmbH  
Zugspitzstr. 4  
D-8011 Vaterstetten
- Zilog Datenbuch der Firma Kontron  
Elektronik GmbH  
Breslauerstr. 2  
8057 Eching bei München
- Z80 CPU-Instruction Set SGS-ATES  
Deutschland Halbleiter  
Bauelemente GmbH  
Haidling 17  
8018 Grafing bei München
- Zilog (U.K. Limited) Nicholson House, Maidenhead, Berkshire U. K.  
Microcomputer 77/78 Zilog Z80 Assemblersprache. Benutzerhand-  
buch in deutscher Sprache. Ausg. April 1978:  
Kontron Elektronik GmbH  
Oskar von Miller Str. 1  
8057 Eching bei München
- Z80 Assembly Language Programing Manual  
Zilog  
10460 Bubb Road  
Cupertino, CA95014  
U.S.A.