256×192

320×256

640×256

$256 \times 192 \times 256$

$320 \times 256 \times 256$

$640 \times 256 \times 6$

Bit	7	6	5	4	3	2	1	0
Function	F	B	P_2	P_1	P_0	I_2	I_1	I_0

Bit	7	6	5	4	3	2	1	0
format 1	P_6	P_5	P_4	P_3	P_2	P_1	P_0	I_0
format 3	P_5	P_4	P_3	P_2	P_1	P_0	I_1	I_0
format 7	P_4	P_3	P_2	P_1	P_0	I_2	I_1	I_0
format 15	P_3	P_2	P_1	P_0	I_3	I_2	I_1	I_0
format 31	P_2	P_1	P_0	I_4	I_3	I_2	I_1	I_0
format 63	P_1	P_0	I_5	I_4	I_3	I_2	I_1	I_0
format 127	P_0	I_6	I_5	I_4	I_3	I_2	I_1	I_0
format 255	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

8×8

32×24

$0R_4R_3Y_2Y_1Y_0R_2R_1R_0C_4C_3C_2C_1C_0$

$$0110 R_4 R_3 R_2 R_1 R_0 C_4 C_3 C_2 C_1 C_0$$

$$1 R_4 R_3 Y_2 Y_1 Y_0 R_2 R_1 R_0 C_4 C_3 C_2 C_1 C_0$$

$$1110 R_4 R_3 R_2 R_1 R_0 C_4 C_3 C_2 C_1 C_0$$

8×1

32×192

512×192

Port 0xff bits 5-3	Attribute	Ink	Paper
000	01111000	black	white
001	01110001	blue	yellow
010	01101010	red	cyan
011	01100011	magenta	green
100	01011100	green	magenta
101	01010101	cyan	red
110	01001110	yellow	blue
111	01000111	white	black

64×32

64×24

$C_0 R_4 R_3 Y_2 Y_1 Y_0 R_2 R_1 R_0 C_5 C_4 C_3 C_2 C_1$

128×96

$100xxxxx$

$640 \times 256 \times 16$

40×32

80×32

40×30

80×30

16×16

$2\times$

$4\times$

$1\times$

4×4

O

Bits 2-0	Slot 1	Slot 2	Slot 3	Slot 4
1	0	1	2	3
3	4	5	6	7
5	4	5	6	3
7	4	7	6	3

Register Group	Register Function Description	Bitma
WR0	Direction Operation and Port A configuration	0XXX
WR1	Port A configuration	0XXX
WR2	Port B configuration	0XXX
WR3	Activation	1XXX
WR5	Ready and Stop configuration	10XX
WR6	Command Register	1XXX

System	Lines	Clocks
48K VGA 50Hz	312	224.0 * 4 = 896
128K VGA 50Hz	311	228.0 * 4 = 912
PENTAGON VGA 50Hz	320	224.0 * 4 = 896
48K VGA 60Hz	262	224.0 * 4 = 896
128K VGA 60Hz	261	228.0 * 4 = 912
HDMI 50Hz	312	216.0 * 4 = 864
HDMI 60Hz	262	214.5 * 4 = 858

System	Lines	Clocks	Freq
48K VGA 50Hz	312	13 977 600	14.0Mhz (28Mhz)
128K VGA 50Hz	311	14 181 600	14.2Mhz (28Mhz)
PENTAGON VGA 50Hz	320	14 336 000	14.3Mhz (28Mhz)
48K VGA 60Hz	262	14 085 120	14.1Mhz (28Mhz)
128K VGA 60Hz	261	14 281 920	14.3Mhz (28Mhz)
HDMI 50Hz	312	13 478 400	13.5Mhz (27Mhz)
HDMI 60Hz	262	13 487 760	13.5Mhz (27Mhz)

System	Max
HDMI	52
Pentagon	54
48k	54
128k	55

clocks/line		slack
858	(52 * 16 = 832)	26 dot clocks
864	(52 * 16 = 832)	32 dot clocks
896	(54 * 16 = 864)	32 dot clocks
912	(55 * 16 = 880)	32 dot clocks

Compare	Standard	Timex	HDMI	48k	128k	Per
0-31	0-255	0-511	Display	Display	Display	Dis
32-36	256-295	512-591	R-Border	R-Border	R-Border	R-I
37	296-303	592-607	R-Border	R-Border	Blanking	Bla
38-48	304-391	608-783	Blanking	Blanking	Blanking	Bla
49	392-399	784-799	L-Border	Blanking	Blanking	L-I
50-52	400-423	800-847	L-Border	L-Border	L-Border	L-I
53-54	424-439	848-879	–	L-Border	L-Border	L-I
55	440-447	880-895	–	–	L-Border	–

– Dot clock compare is out of range.

Line	HDMI 50Hz	HDMI 60Hz	48k 50Hz	48k 60Hz	128k 50Hz	128k 60Hz	Pentagon
0-191	Display	Display	Display	Display	Display	Display	Display
192-211	B-Border	B-Border	B-Border	B-Border	B-Border	B-Border	B-Border
212-224	B-Border	Blanking	B-Border	B-Border	B-Border	B-Border	B-Border
225-231	B-Border	Blanking	B-Border	Blanking	B-Border	Blanking	B-Border
232-238	Blanking	Blanking	B-Border	Blanking	B-Border	Blanking	B-Border
239	Blanking	Blanking	B-Border	T-Border	B-Border	T-Border	B-Border
240	Blanking	Blanking	B-Border	T-Border	B-Border	T-Border	B-Border
241-244	Blanking	Blanking	B-Border	T-Border	B-Border	T-Border	Blanking
245-247	Blanking	T-Border	B-Border	T-Border	B-Border	T-Border	Blanking
248	Blanking	T-Border	B-Border*	T-Border	B-Border*	T-Border	Blanking
249-255	Blanking	T-Border	Blanking	T-Border	Blanking	T-Border	Blanking
255	Blanking	T-Border	Blanking	T-Border	Blanking	T-Border	T-Border
256	Blanking*	T-Border	Blanking	T-Border	Blanking	T-Border	T-Border
257-260	Blanking	T-Border	Blanking	T-Border	Blanking	T-Border	T-Border
261	Blanking	T-Border	Blanking	T-Border	Blanking	–	T-Border
262	Blanking	–	Blanking	–	Blanking	–	T-Border
263-271	Blanking	–	T-Border	–	T-Border	–	T-Border
272-310	T-Border	–	T-Border	–	T-Border	–	T-Border
311	T-Border	–	T-Border	–	–	–	T-Border
312-319	–	–	–	–	–	–	T-Border

– Line compare is out of range

* ULA VBLANK interrupt.

352×288

352×240

Freq	Resolution	Top	Bottom	Left	Right
50Hz	336x288	32	32	40	40
60Hz	336x240	24	24	40	40

Timing	Video	Ref	Lines	Top	Bot	Left	Right	Ext	Re
0/1 48k	VGA	50Hz	312	280	223	51.1	36.15	80x64	33
0/1 48k	VGA	60Hz	262	246	207	51.1	36.15	80x48	33
2/3 128k	VGA	50Hz	311	279	223	52.1	36.15	80x64	33
2/3 128k	VGA	60Hz	261	245	207	52.1	36.15	80x48	33
4 Pentagon	VGA	50Hz	320	288	223	51.1	36.15	80x64	33
0/1 48k	HDMI	50Hz	312	280	223	49.1	36.15	80x64	33
0/1 48k	HDMI	60Hz	262	246	207	48.11	36.15	80x48	33
2/3 128k	HDMI	50Hz	312	280	223	49.1	36.15	80x64	33
2/3 128k	HDMI	60Hz	262	246	207	48.11	36.15	80x48	33
4 Pentagon	HDMI	50Hz	312	280	223	49.1	36.15	80x64	33
4 Pentagon	HDMI	60Hz	262	246	207	48.11	36.15	80x48	33

TOP: Initial line of the extended top border area - see notes below*
BOT: Last line of the extended bottom border area - see notes below*
LEFT: First pixel of the extended left border area - see notes below**
RIGHT: Last pixel of the extended right border area - see notes below**
* Line compare value for MOVE (bits 8..0).
** The integer part is the horizontal value for MOVE (bits 14..9).
** The fractional part is specified in dot clocks (NOOP instructions).

Name	15-8	7-0	Clocks
NOOP	00000000	00000000	1
MOVE	0RRRRRRR	DDDDDDDD	2
WAIT	1HHHHHHV	VVVVVVVV	1

H 6 bit horizontal dot clock compare

V 9 bit vertical line compare

R 7 bit Next register 0x00..0x7F

D 8 bit data

Reg	7-0	Description
0x60	DDDDDDDD	BYTE data to write to COPPER program RAM
0x61	IIIIIII	Program RAM index 7..0
0x62	CC000III	Program RAM index 10..8 and control bits

D 8 bit data

I 11 bit index

C 2 bit control

Name	CC	Description
STOP	00	STOP COPPER
RESET	01	RESET PC and start COPPER
START	10	START COPPER

* The control mode names used in this guide differ from the official names.

Timing	Video	Refresh	T-States	Clocks	Lines	Width	HI
0/1 48k	VGA	50Hz	224	896	312	256	44
0/1 48k	VGA	50Hz	224	896	312	512	44
0/1 48k	VGA	60Hz	224	896	262	256	44
0/1 48k	VGA	60Hz	224	896	262	512	44
2/3 128k	VGA	50Hz	228	912	311	256	45
2/3 128k	VGA	50Hz	228	912	311	512	45
2/3 128k	VGA	60Hz	228	912	261	256	45
2/3 128k	VGA	60Hz	228	912	261	512	45
4 Pentagon	VGA	50Hz	224	896	320	256	44
4 Pentagon	VGA	50Hz	224	896	320	512	44
0/1 48k	HDMI	50Hz	216	864	312	256	43
0/1 48k	HDMI	50Hz	216	864	312	512	43
0/1 48k	HDMI	60Hz	214.5	858	262	256	42
0/1 48k	HDMI	60Hz	214.5	858	262	512	42
2/3 128k	HDMI	50Hz	216	864	312	256	43
2/3 128k	HDMI	50Hz	216	864	312	512	43
2/3 128k	HDMI	60Hz	214.5	858	262	256	43
2/3 128k	HDMI	60Hz	214.5	858	262	512	43
4 Pentagon	HDMI	50Hz	216	864	312	256	43
4 Pentagon	HDMI	50Hz	216	864	312	512	43
4 Pentagon	HDMI	60Hz	214.5	858	262	256	43
4 Pentagon	HDMI	60Hz	214.5	858	262	512	43

Syntax: **COLOURS [RGB]** ***paper ink***

paper Paper (background) colour

ink	Ink (foreground) colour

RGB	Causes ink and paper colours to be interpreted as 9-bit octal RGB numbers

Syntax: COPYSYS

Syntax: **DATE**
DATE C
DATE CONTINUOUS
DATE *time-specification*
DATE SET

time-specification Time/date in the format MM/DD/YY HH:MM:SS

C Continuously show the date and time until a key is pressed

CONTINUOUS Continuously show the date and time until a key is pressed

SET Prompt the user for the current date and time

Offset-7	Description
00 - 04	Reserved For System Use
05	BDOS version number
06 - 09	User Flags
0A - 0F	Reserved For System Use
10 - 11	Program Error return code
12 - 19	Reserved For System Use
1A	Console Width (columns)
1B	Console Column Position
1C	Console Page Length
1D - 21	Reserved For System Use
22 - 23	CONIN Redirection flag
24 - 25	CONOUT Redirection flag
26 - 27	AUXIN Redirection flag
2A - 2B	LSTOUT Redirection flag
2C	Page Mode
2D	Reserved For System Use
2E	CTRL-H Active
2F	Rubout Active
30 - 32	Reserved For System Use
33 - 34	Console Mode
35 - 36	Reserved For System Use
37	Output Delimiter
39 - 3B	Reserved For System Use
3C - 3D	Current DMA Address
3E	Current Disk
3F - 43	Reserved For System Use
44	Current User Number
45 - 49	Reserved For System Use
4A	BDOS Multi-Sector Count
4B	BDOS Error Mode
4C - 4F	Drive Search Chain (DISKS A:E:F:)
50	Temporary File Drive
51	Error Disk
52 - 56	Reserved For System Use
57	BDOS flags
58 - 5C	Date Stamp
5D - 5E	Common Memory Base Address
5F - 63	Reserved For System Use

Code	Meaning
0000 - FEFF	Successful return
FF00 - FFFE	Unsuccessful return
0000	The CCP initializes the Program Return Code to zero u
	loaded as the result of program chain.
FF00 - FFFC	Reserved
FFFD	The program is terminated because of a fatal BDOS err
FFFE	The program is terminated by the BDOS because the u

Location	Contents
byte 0	The drive field is set to the specified drive. If the drive is not specified the default drive code is used. 0 = default 1 = A 2 = B.
byte 1-8	The name is set to the specified filename. All letters are converted to upper-case. If the name is not eight chara long the remaining bytes in the filename field are padded w blanks. If the filename has an asterisk * all remaining byt in the filename field are filled in with question marks ?. A error occurs if the filename is more than eight bytes long.
byte 9-11	The type is set to the specified filetype. If no filetype is specified the type field is initialized to blanks. All letters are converted to upper-case. If the type not three characters long the remaining bytes in the filetyp field are padded with blanks. If an asterisk * occurs all remaining bytes are filled in with question marks ?. An er occurs if the type field is more than three bytes long.
byte 12-15	Filled in with zeros.
byte 16-23	The password field is set to the specified password. If no password is specified it is initialized to blanks. If the password is less than eight characters long remaining bytes are padded with blanks. All letters are co to upper-case. If the password field is more than eight byte long an error occurs. Note that a blank in the first positio the password field implies no password was specified.
byte 24-31	Reserved for system use.

R	W	16————————-0	Port(hex)	Description
*	*	XXXX XXXX XXXX XXX0	$fe	ULA
*	*	XXXX XXXX 1111 1111	$ff	Timex video/floating bus
	*	0XXX XXXX XXXX XX01	$7ffd	ZX Spectrum 128 memory
	*	01XX XXXX XXXX XX01	$7ffd	ZX Spectrum 128 memory +3 only
	*	1101 XXXX XXXX XX01	$dffd	ZX Spectrum 128 memory (precedence over AY)
	*	0001 XXXX XXXX XX01	$1ffd	ZX Spectrum +3 memory
*		0000 XXXX XXXX XX01		ZX Spectrum +3 floating bus
*	*	0010 0100 0011 1011	$243b	NextREG Register Select
*	*	0010 0101 0011 1011	$253b	NextREG data/value
	*	0001 0000 0011 1011	$103b	i2c SCL (rtc)
*	*	0001 0001 0011 1011	$113b	i2c SDA (rtc)
*	*	0001 0010 0011 1011	$123b	Layer 2
*	*	0001 0011 0011 1011	$133b	UART tx
*	*	0001 0100 0011 1011	$143b	UART rx
*	*	0001 0101 0011 1011	$153b	UART control
*	*	XXXX XXXX 0110 1011	$6b	zxnDMA
*	*	11XX XXXX XXXX X101	$fffd	AY reg
	*	10XX XXXX XXXX X101	$bffd	AY dat
	*	XXXX XXXX 0000 1111	$0f	DAC A
	*	XXXX XXXX 1111 0001	$f1	DAC A (precedence over XXFD)
	*	XXXX XXXX 0011 1111	$3f	DAC A
	*	XXXX XXXX 1101 1111	$df	DAC A/C specdrum
	*	XXXX XXXX 0001 1111	$1f	DAC B
	*	XXXX XXXX 1111 0011	$f3	DAC B
	*	XXXX XXXX 0100 1111	$4f	DAC C
	*	XXXX XXXX 1111 1001	$f9	DAC C (precedence over XXFD)
	*	XXXX XXXX 0101 1111	$5f	DAC D
	*	XXXX XXXX 1111 1011	$fb	DAC D
	*	XXXX XXXX 1110 0111	$e7	SPI /CS (sd card/flash/rpi)
*	*	XXXX XXXX 1110 1011	$eb	SPI /DATA
*	*	XXXX XXXX 1110 0011	$e3	divMMC Control
*		XXXX 1011 1101 1111	$fbdf	Kempston mouse x
*		XXXX 1111 1101 1111	$ffdf	Kempston mouse y
*		XXXX 1010 1101 1111	$fadf	Kempston mouse wheel/buttons
*		XXXX XXXX 0001 1111	$1f	Kempston joy 1
*		XXXX XXXX 0011 0111	$37	Kempston joy 2
*	*	XXXX XXXX 0001 1111	$1f	Multiface 1 disable
*		XXXX XXXX 1001 1111	$9f	Multiface 1 enable
*	*	XXXX XXXX 0011 1111	$3f	Multiface 128 disable
*		XXXX XXXX 1011 1111	$bf	Multiface 128 enable
*	*	XXXX XXXX 1011 1111	$bf	Multiface +3 disable
*		XXXX XXXX 0011 1111	$3f	Multiface +3 enable
*	*	0011 0000 0011 1011	$303b	Sprite slot/flags
	*	XXXX XXXX 0101 0111	$57	Sprite attributes
	*	XXXX XXXX 0101 1011	$5b	Sprite pattern
	*	1011 1111 0011 1011	$bf3b	ULAPlus register
*	*	1111 1111 0011 1011	$ff3b	ULAPlus data


CS

DATA


RESET

IORQULA

ROMCS

$$\text{Divider} = \frac{538461}{\text{Rate}} - 1$$

$$\text{Rate} = \frac{538461}{\text{Divider}+1}$$

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$00	nop	nop	1	4	$10	djnz x	–	2	13/8
$01	ld bc,xx	lxi b,xx	3	10	$11	ld de,xx	lxi d,xx	3	10
$02	ld (bc),a	stax b	1	7	$12	ld (de),a	stax d	1	7
$03	inc bc	inx b	1	6	$13	inc de	inx d	1	6
$04	inc b	inr b	1	4	$14	inc d	inr d	1	4
$05	dec b	dcr b	1	4	$15	dec d	dcr d	1	4
$06	ld b,x	mvi b,x	2	7	$16	ld d,x	mvi d,x	2	7
$07	rlca	rlc	1	4	$17	rla	ral	1	4
$08	ex af,af’	–	1	4	$18	jr x	–	2	12
$09	add hl,bc	dad b	1	11	$19	add hl,de	dad d	1	11
$0A	ld a,(bc)	ldax b	1	7	$1A	ld a,(de)	ldax d	1	7
$0B	dec bc	dcx b	1	6	$1B	dec de	dcx d	1	6
$0C	inc c	icr c	1	4	$1C	inc e	icr e	1	4
$0D	dec c	dcr c	1	4	$1D	dec e	dcr e	1	4
$0E	ld c,x	mvi c,x	2	7	$1E	ld e,x	mvi e,x	2	7
$0F	rrca	rrc	1	4	$1F	rra	rar	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$20	jr nz,x	–	2	12/7	$30	jr nc,x	–	2	12/7
$21	ld hl,xx	lxi h,xx	3	10	$31	ld sp,xx	lxi sp,xx	3	10
$22	ld (xx),hl	shld xx	3	16	$32	ld (xx),a	sta xx	3	13
$23	inc hl	inx h	1	6	$33	inc sp	inx sp	1	6
$24	inc h	inr h	1	4	$34	inc (hl)	inr m	1	11
$25	dec h	dcr h	1	4	$35	dec (hl)	dcr m	1	11
$26	ld h,x	mvi h,x	2	7	$36	ld (hl),x	mvi m,x	2	10
$27	daa	daa	1	4	$37	scf	stc	1	4
$28	jr z,x	–	2	12/7	$38	jr c,x	–	2	12/7
$29	add hl,hl	dad h	1	11	$39	add hl,sp	dad sp	1	11
$2A	ld hl,(xx)	lhld xx	3	16	$3A	ld a,(xx)	lda xx	3	13
$2B	dec hl	dcx h	1	6	$3B	dec sp	dcx sp	1	6
$2C	inc l	inr l	1	4	$3C	inc a	inr a	1	4
$2D	dec l	dcr l	1	4	$3D	dec a	dcr a	1	4
$2E	ld l,x	mvi l,x	2	7	$3E	ld a,x	mvi a,x	2	7
$2F	cpl	cma	1	4	$3F	ccf	cmc	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$40	ld b,b	mov b,b	1	4	$50	ld d,b	mov d,b	1	4
$41	ld b,c	mov b,c	1	4	$51	ld d,c	mov d,c	1	4
$42	ld b,d	mov b,d	1	4	$52	ld d,d	mov d,d	1	4
$43	ld b,e	mov b,e	1	4	$53	ld d,e	mov d,e	1	4
$44	ld b,h	mov b,h	1	4	$54	ld d,h	mov d,h	1	4
$45	ld b,l	mov b,l	1	4	$55	ld d,l	mov d,l	1	4
$46	ld b,(hl)	mov b,m	1	7	$56	ld d,(hl)	mov d,m	1	7
$47	ld b,a	mov b,a	1	4	$57	ld d,a	mov d,a	1	4
$48	ld c,b	mov c,b	1	4	$58	ld e,b	mov e,b	1	4
$49	ld c,c	mov c,c	1	4	$59	ld e,c	mov e,c	1	4
$4A	ld c,d	mov c,d	1	4	$5A	ld e,d	mov e,d	1	4
$4B	ld c,e	mov c,e	1	4	$5B	ld e,e	mov e,e	1	4
$4C	ld c,h	mov c,h	1	4	$5C	ld e,h	mov e,h	1	4
$4D	ld c,l	mov c,l	1	4	$5D	ld e,l	mov e,l	1	4
$4E	ld c,(hl)	mov c,m	1	7	$5E	ld e,(hl)	mov e,m	1	7
$4F	ld c,a	mov c,a	1	4	$5F	ld e,a	mov e,a	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$60	ld h,b	mov h,b	1	4	$70	ld (hl),b	mov m,b	1	4
$61	ld h,c	mov h,c	1	4	$71	ld (hl),c	mov m,c	1	4
$62	ld h,d	mov h,d	1	4	$72	ld (hl),d	mov m,d	1	4
$63	ld h,e	mov h,e	1	4	$73	ld (hl),e	mov m,e	1	4
$64	ld h,h	mov h,h	1	4	$74	ld (hl),h	mov m,h	1	4
$65	ld h,l	mov h,l	1	4	$75	ld (hl),l	mov m,l	1	4
$66	ld h,(hl)	mov h,m	1	7	$76	halt	halt	1	4+
$67	ld h,a	mov h,a	1	4	$77	ld (hl),a	mov m,a	1	7
$68	ld l,b	mov l,b	1	4	$78	ld a,b	mov a,b	1	4
$69	ld l,c	mov l,c	1	4	$79	ld a,c	mov a,c	1	4
$6A	ld l,d	mov l,d	1	4	$7A	ld a,d	mov a,d	1	4
$6B	ld l,e	mov l,e	1	4	$7B	ld a,e	mov a,e	1	4
$6C	ld l,h	mov l,h	1	4	$7C	ld a,h	mov a,h	1	4
$6D	ld l,l	mov l,l	1	4	$7D	ld a,l	mov a,l	1	4
$6E	ld l,(hl)	mov l,m	1	7	$7E	ld a,(hl)	mov a,m	1	7
$6F	ld l,a	mov l,a	1	4	$7F	ld a,a	mov a,a	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$80	add a,b	add b	1	4	$90	sub b	sub b	1	4
$81	add a,c	add c	1	4	$91	sub c	sub c	1	4
$82	add a,d	add d	1	4	$92	sub d	sub d	1	4
$83	add a,e	add e	1	4	$93	sub e	sub e	1	4
$84	add a,h	add h	1	4	$94	sub h	sub h	1	4
$85	add a,l	add l	1	4	$95	sub l	sub l	1	4
$86	add a,(hl)	add m	1	7	$96	sub (hl)	sub m	1	7
$87	add a,a	add a	1	4	$97	sub a	sub a	1	4
$88	adc a,b	adc b	1	4	$98	sbc a,b	sbb b	1	4
$89	adc a,c	adc c	1	4	$99	sbc a,c	sbb c	1	4
$8A	adc a,d	adc d	1	4	$9A	sbc a,d	sbb d	1	4
$8B	adc a,e	adc e	1	4	$9B	sbc a,e	sbb e	1	4
$8C	adc a,h	adc h	1	4	$9C	sbc a,h	sbb h	1	4
$8D	adc a,l	adc l	1	4	$9D	sbc a,l	sbb l	1	4
$8E	adc a,(hl)	adc m	1	7	$9E	sbc a,(hl)	sbb m	1	7
$8F	adc a,a	adc a	1	4	$9F	sbc a,a	sbb a	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$A0	and b	ana b	1	4	$B0	or b	ora b	1	4
$A1	and c	ana c	1	4	$B1	or c	ora c	1	4
$A2	and d	ana d	1	4	$B2	or d	ora d	1	4
$A3	and e	ana e	1	4	$B3	or e	ora e	1	4
$A4	and h	ana h	1	4	$B4	or h	ora h	1	4
$A5	and l	ana l	1	4	$B5	or l	ora l	1	4
$A6	and (hl)	ana m	1	7	$B6	or (hl)	ora m	1	7
$A7	and a	ana a	1	4	$B7	or a	ora a	1	4
$A8	xor b	xra b	1	4	$B8	cp b	cmp b	1	4
$A9	xor c	xra c	1	4	$B9	cp c	cmp c	1	4
$AA	xor d	xra d	1	4	$BA	cp d	cmp d	1	4
$AB	xor e	xra e	1	4	$BB	cp e	cmp e	1	4
$AC	xor h	xra h	1	4	$BC	cp h	cmp h	1	4
$AD	xor l	xra l	1	4	$BD	cp l	cmp l	1	4
$AE	xor (hl)	xra m	1	7	$BE	cp (hl)	cmp m	1	7
$AF	xor a	xra a	1	4	$BF	cp a	cmp a	1	4

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$C0	ret nz	rnz	1	11/5	$D0	ret nc	rnc	1	11/5
$C1	pop bc	pop b	1	10	$D1	pop de	pop d	1	10
$C2	jp nz,xx	jnz xx	3	10	$D2	jp nc,xx	jnc xx	3	10
$C3	jp xx	jmp xx	3	10	$D3	out (x),a	out x	2	11
$C4	call nz,xx	cnz xx	3	17/10	$D4	call nc,xx	cnc xx	3	17/10
$C5	push bc	push b	1	11	$D5	push de	push d	1	11
$C6	add a,x	adi x	2	7	$D6	sub x	sui x	2	7
$C7	rst 00h	rst 0	1	11	$D7	rst 10h	rst 2	1	11
$C8	ret z	rz	1	11/5	$D8	ret c	rc	1	11/5
$C9	ret	ret	1	10	$D9	exx	–	1	4
$CA	jp z,xx	jz xx	3	10	$DA	jp c,xx	jc xx	3	10
$CB	xxBITxx	–	+1	–	$DB	in a,(x)	in x	2	11
$CC	call z,xx	cz xx	3	17/10	$DC	call c,xx	cc xx	3	17/11
$CD	call xx	call xx	3	17	$DD	xxIXxx	–	+1	–
$CE	adc a,x	aci x	2	7	$DE	sbc a,x	sbi x	2	7
$CF	rst 08h	rst 1	1	11	$DF	rst 18h	rst 3	1	11

Op	Z80	8080	Sz	T	Op	Z80	8080	Sz	T
$E0	ret po	rpo	1	11/5	$F0	ret p	rp	1	11/5
$E1	pop hl	pop h	1	10	$F1	pop af	pop psw	1	10
$E2	jp po,xx	jpo xx	3	10	$F2	jp p,xx	jp xx	3	10
$E3	ex (sp),hl	xthl	1	19	$F3	di	di	1	4
$E4	call po,xx	cpo xx	3	17/10	$F4	call p,xx	cp xx	3	17/10
$E5	push hl	push h	1	11	$F5	push af	push psw	1	11
$E6	and x	ani x	2	7	$F6	or x	ori x	2	7
$E7	rst 20h	rst 4	1	11	$F7	rst 30h	rst 6	1	11
$E8	ret pe	rpe	1	11/5	$F8	ret m	rm	1	11/5
$E9	jp (hl)	pchl	1	4	$F9	ld sp,hl	sphl	1	6
$EA	jp pe,xx	jpe xx	3	10	$FA	jp m,xx	jm xx	3	10
$EB	ex de,hl	xchg	1	4	$FB	ei	ei	1	4
$EC	call pe,xx	cpe	3	17/10	$FC	call m,xx	cm xx	3	17/10
$ED	xx80xx	–	+1	–	$FD	xxIYxx	–	+1	–
$EE	xor x	xri x	2	7	$FE	cp x	cpi x	2	7
$EF	rst 28h	rst 5	1	11	$FF	rst 38h	rst 7	1	11

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CB00	rlc b	2	8	$CB10	rl b	2	8
$CB01	rlc c	2	8	$CB11	rl c	2	8
$CB02	rlc d	2	8	$CB12	rl d	2	8
$CB03	rlc e	2	8	$CB13	rl e	2	8
$CB04	rlc h	2	8	$CB14	rl h	2	8
$CB05	rlc l	2	8	$CB15	rl l	2	8
$CB06	rlc (hl)	2	15	$CB16	rl (hl)	2	15
$CB07	rlc a	2	8	$CB17	rl a	2	8
$CB08	rrc b	2	8	$CB18	rr b	2	8
$CB09	rrc c	2	8	$CB19	rr c	2	8
$CB0A	rrc d	2	8	$CB1A	rr d	2	8
$CB0B	rrc e	2	8	$CB1B	rr e	2	8
$CB0C	rrc h	2	8	$CB1C	rr h	2	8
$CB0D	rrc l	2	8	$CB1D	rr l	2	8
$CB0E	rrc (hl)	2	15	$CB1E	rr (hl)	2	15
$CB0F	rrc a	2	8	$CB1F	rr a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CB20	sla b	2	8	$CB30	sll b	2	8
$CB21	sla c	2	8	$CB31	sll c	2	8
$CB22	sla d	2	8	$CB32	sll d	2	8
$CB23	sla e	2	8	$CB33	sll e	2	8
$CB24	sla h	2	8	$CB34	sll h	2	8
$CB25	sla l	2	8	$CB35	sll l	2	8
$CB26	sla (hl)	2	15	$CB36	sll (hl)	2	15
$CB27	sla a	2	8	$CB37	sll a	2	8
$CB28	sra b	2	8	$CB38	srl b	2	8
$CB29	sra c	2	8	$CB39	srl c	2	8
$CB2A	sra d	2	8	$CB3A	srl d	2	8
$CB2B	sra e	2	8	$CB3B	srl e	2	8
$CB2C	sra h	2	8	$CB3C	srl h	2	8
$CB2D	sra l	2	8	$CB3D	srl l	2	8
$CB2E	sra (hl)	2	15	$CB3E	srl (hl)	2	15
$CB2F	sra a	2	8	$CB3F	srl a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CB40	bit 0,b	2	8	$CB50	bit 2,b	2	8
$CB41	bit 0,c	2	8	$CB51	bit 2,c	2	8
$CB42	bit 0,d	2	8	$CB52	bit 2,d	2	8
$CB43	bit 0,e	2	8	$CB53	bit 2,e	2	8
$CB44	bit 0,h	2	8	$CB54	bit 2,h	2	8
$CB45	bit 0,l	2	8	$CB55	bit 2,l	2	8
$CB46	bit 0,(hl)	2	12	$CB56	bit 2,(hl)	2	12
$CB47	bit 0,a	2	8	$CB57	bit 2,a	2	8
$CB48	bit 1,b	2	8	$CB58	bit 3,b	2	8
$CB49	bit 1,c	2	8	$CB59	bit 3,c	2	8
$CB4A	bit 1,d	2	8	$CB5A	bit 3,d	2	8
$CB4B	bit 1,e	2	8	$CB5B	bit 3,e	2	8
$CB4C	bit 1,h	2	8	$CB5C	bit 3,h	2	8
$CB4D	bit 1,l	2	8	$CB5D	bit 3,l	2	8
$CB4E	bit 1,(hl)	2	12	$CB5E	bit 3,(hl)	2	12
$CB4F	bit 1,a	2	8	$CB5F	bit 3,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CB60	bit 4,b	2	8	$CB70	bit 6,b	2	8
$CB61	bit 4,c	2	8	$CB71	bit 6,c	2	8
$CB62	bit 4,d	2	8	$CB72	bit 6,d	2	8
$CB63	bit 4,e	2	8	$CB73	bit 6,e	2	8
$CB64	bit 4,h	2	8	$CB74	bit 6,h	2	8
$CB65	bit 4,l	2	8	$CB75	bit 6,l	2	8
$CB66	bit 4,(hl)	2	12	$CB76	bit 6,(hl)	2	12
$CB67	bit 4,a	2	8	$CB77	bit 6,a	2	8
$CB68	bit 5,b	2	8	$CB78	bit 7,b	2	8
$CB69	bit 5,c	2	8	$CB79	bit 7,c	2	8
$CB6A	bit 5,d	2	8	$CB7A	bit 7,d	2	8
$CB6B	bit 5,e	2	8	$CB7B	bit 7,e	2	8
$CB6C	bit 5,h	2	8	$CB7C	bit 7,h	2	8
$CB6D	bit 5,l	2	8	$CB7D	bit 7,l	2	8
$CB6E	bit 5,(hl)	2	12	$CB7E	bit 7,(hl)	2	12
$CB6F	bit 5,a	2	8	$CB7F	bit 7,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CB80	res 0,b	2	8	$CB90	res 2,b	2	8
$CB81	res 0,c	2	8	$CB91	res 2,c	2	8
$CB82	res 0,d	2	8	$CB92	res 2,d	2	8
$CB83	res 0,e	2	8	$CB93	res 2,e	2	8
$CB84	res 0,h	2	8	$CB94	res 2,h	2	8
$CB85	res 0,l	2	8	$CB95	res 2,l	2	8
$CB86	res 0,(hl)	2	15	$CB96	res 2,(hl)	2	15
$CB87	res 0,a	2	8	$CB97	res 2,a	2	8
$CB88	res 1,b	2	8	$CB98	res 3,b	2	8
$CB89	res 1,c	2	8	$CB99	res 3,c	2	8
$CB8A	res 1,d	2	8	$CB9A	res 3,d	2	8
$CB8B	res 1,e	2	8	$CB9B	res 3,e	2	8
$CB8C	res 1,h	2	8	$CB9C	res 3,h	2	8
$CB8D	res 1,l	2	8	$CB9D	res 3,l	2	8
$CB8E	res 1,(hl)	2	15	$CB9E	res 3,(hl)	2	15
$CB8F	res 1,a	2	8	$CB9F	res 3,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CBA0	res 4,b	2	8	$CBB0	res 6,b	2	8
$CBA1	res 4,c	2	8	$CBB1	res 6,c	2	8
$CBA2	res 4,d	2	8	$CBB2	res 6,d	2	8
$CBA3	res 4,e	2	8	$CBB3	res 6,e	2	8
$CBA4	res 4,h	2	8	$CBB4	res 6,h	2	8
$CBA5	res 4,l	2	8	$CBB5	res 6,l	2	8
$CBA6	res 4,(hl)	2	15	$CBB6	res 6,(hl)	2	15
$CBA7	res 4,a	2	8	$CBB7	res 6,a	2	8
$CBA8	res 5,b	2	8	$CBB8	res 7,b	2	8
$CBA9	res 5,c	2	8	$CBB9	res 7,c	2	8
$CBAA	res 5,d	2	8	$CBBA	res 7,d	2	8
$CBAB	res 5,e	2	8	$CBBB	res 7,e	2	8
$CBAC	res 5,h	2	8	$CBBC	res 7,h	2	8
$CBAD	res 5,l	2	8	$CBBD	res 7,l	2	8
$CBAE	res 5,(hl)	2	15	$CBBE	res 7,(hl)	2	15
$CBAF	res 5,a	2	8	$CBBF	res 7,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CBC0	set 0,b	2	8	$CBD0	set 2,b	2	8
$CBC1	set 0,c	2	8	$CBD1	set 2,c	2	8
$CBC2	set 0,d	2	8	$CBD2	set 2,d	2	8
$CBC3	set 0,e	2	8	$CBD3	set 2,e	2	8
$CBC4	set 0,h	2	8	$CBD4	set 2,h	2	8
$CBC5	set 0,l	2	8	$CBD5	set 2,l	2	8
$CBC6	set 0,(hl)	2	15	$CBD6	set 2,(hl)	2	15
$CBC7	set 0,a	2	8	$CBD7	set 2,a	2	8
$CBC8	set 1,b	2	8	$CBD8	set 3,b	2	8
$CBC9	set 1,c	2	8	$CBD9	set 3,c	2	8
$CBCA	set 1,d	2	8	$CBDA	set 3,d	2	8
$CBCB	set 1,e	2	8	$CBDB	set 3,e	2	8
$CBCC	set 1,h	2	8	$CBDC	set 3,h	2	8
$CBCD	set 1,l	2	8	$CBDD	set 3,l	2	8
$CBCE	set 1,(hl)	2	15	$CBDE	set 3,(hl)	2	15
$CBCF	set 1,a	2	8	$CBDF	set 3,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$CBE0	set 4,b	2	8	$CBF0	set 6,b	2	8
$CBE1	set 4,c	2	8	$CBF1	set 6,c	2	8
$CBE2	set 4,d	2	8	$CBF2	set 6,d	2	8
$CBE3	set 4,e	2	8	$CBF3	set 6,e	2	8
$CBE4	set 4,h	2	8	$CBF4	set 6,h	2	8
$CBE5	set 4,l	2	8	$CBF5	set 6,l	2	8
$CBE6	set 4,(hl)	2	15	$CBF6	set 6,(hl)	2	15
$CBE7	set 4,a	2	8	$CBF7	set 6,a	2	8
$CBE8	set 5,b	2	8	$CBF8	set 7,b	2	8
$CBE9	set 5,c	2	8	$CBF9	set 7,c	2	8
$CBEA	set 5,d	2	8	$CBFA	set 7,d	2	8
$CBEB	set 5,e	2	8	$CBFB	set 7,e	2	8
$CBEC	set 5,h	2	8	$CBFC	set 7,h	2	8
$CBED	set 5,l	2	8	$CBFD	set 7,l	2	8
$CBEE	set 5,(hl)	2	15	$CBFE	set 7,(hl)	2	15
$CBEF	set 5,a	2	8	$CBFF	set 7,a	2	8

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$DD09	add ix,bc	2	15	$DD35	dec (ix+x)	3	23
$DD19	add ix,de	2	15	$DD36	ld (ix+x),x	5	19
$DD21	ld ix,xx	4	14	$DD39	add ix,sp	2	15
$DD22	ld (xx),ix	4	20	$DD44	ld b,ixh	2	8
$DD23	inc ix	2	10	$DD45	ld b,ixl	2	8
$DD24	inc ixh	2	8	$DD46	ld b,(ix+x)	2	19
$DD25	dec ixh	2	8	$DD4C	ld c,ixh	2	8
$DD26	ld ixh,x	3	11	$DD4D	ld c,ixl	2	8
$DD29	add ix,ix	2	15	$DD4E	ld c,(ix+x)	3	19
$DD2A	ld ix,(xx)	4	20	$DD54	ld d,ixh	2	8
$DD2B	dec ix	2	10	$DD55	ld d,ixl	2	8
$DD2C	inc ixl	2	8	$DD56	ld d,(ix+x)	3	19
$DD2D	dec ixl	2	8	$DD5C	ld e,ixh	2	8
$DD2E	ld ixl,x	4	11	$DD5D	ld e,ixl	2	8
$DD34	inc (ix+x)	3	23	$DD5E	ld e,(ix+x)	3	19

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$DD60	ld ixh,b	2	8	$DD70	ld (ix+x),b	3	19
$DD61	ld ixh,c	2	8	$DD71	ld (ix+x),c	3	19
$DD62	ld ixh,d	2	8	$DD72	ld (ix+x),d	3	19
$DD63	ld ixh,e	2	8	$DD73	ld (ix+x),e	3	19
$DD64	ld ixh,ixh	2	8	$DD74	ld (ix+x),h	3	19
$DD65	ld h,(ix+x)	3	19	$DD75	ld (ix+x),l	3	19
$DD65	ld ixh,ixl	2	8	$DD77	ld (ix+x),a	3	19
$DD67	ld ixh,a	2	8	$DD7C	ld a,ixh	2	8
$DD68	ld ixl,b	2	8	$DD7D	ld a,ixl	2	8
$DD69	ld ixl,c	2	8	$DD7E	ld a,(ix+x)	3	19
$DD6A	ld ixl,d	2	8	$DD84	add a,ixh	2	8
$DD6B	ld ixl,e	2	8	$DD85	add a,ixl	2	8
$DD6C	ld ixl,ixh	2	2	$DD86	add a,(ix+x)	3	19
$DD6D	ld ixl,ixl	2	2	$DD8C	adc a,ixh	2	8
$DD6E	ld l,(ix+x)	3	19	$DD8D	adc a,ixl	2	8
$DD6F	ld ixl,a	2	8	$DD8E	adc a,(ix+x)	3	19

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$DD94	sub ixh	2	8	$DDB4	or ixh	2	8
$DD95	sub ixl	2	8	$DDB5	or ixl	2	8
$DD96	sub (ix+x)	3	19	$DDB6	or (ix+x)	3	19
$DD9C	sbc a,ixh	2	8	$DDBC	cp ixh	2	8
$DD9D	sbc a,ixl	2	8	$DDBD	cp ixl	2	8
$DD9E	sbc a,(ix+x)	3	1	$DDBE	cp (ix+x)	2	19
$DDA4	and ixh	2	8	$DDCB	xBIT+IXx	+1	–
$DDA5	and ixl	2	8	$DDE1	pop ix	2	14
$DDA6	and (ix+x)	3	19	$DDE3	ex (sp),ix	2	23
$DDAC	xor ixh	2	8	$DDE5	push ix	2	15
$DDAD	xor ixl	2	8	$DDE9	jp (ix)	3	8
$DDAE	xor (ix+x)	3	19	$DDF9	ld sp,ix	2	10

Opcode	Mnemonic	Bytes	Timing	Opcode	Mnemonic	Bytes	Timing
$ED23	swapinb *	2	8	$ED40	in b,(c)	2	12
$ED24	mirror a *	2	8	$ED41	out (c),b	2	12
$ED27	test x *	3	11	$ED42	sbc hl,bc	2	15
$ED28	bsla de,b *	2	8	$ED43	ld (xx),bc	4	20
$ED29	bsra de,b *	2	8	$ED44	neg	2	8
$ED2A	bsrl de,b *	2	8	$ED45	retn	2	14
$ED2B	bsrf de,b *	2	8	$ED46	im 0	2	8
$ED2C	brlc de,b *	2	8	$ED47	ld i,a	2	9
$ED30	mul d,e *	2	8	$ED48	in c,(c)	2	12
$ED31	add hl,a *	2	8	$ED49	out (c),c	2	12
$ED32	add de,a *	2	8	$ED4A	adc hl,bc	2	15
$ED33	add bc,a *	2	8	$ED4B	ld bc,(xx)	4	20
$ED34	add hl,xx *	4	16	$ED4D	reti	2	14
$ED35	add de,xx *	4	16	$ED4F	ld r,a	2	9
$ED36	add bc,xx *	4	16				

* ZX Spectrum Next extension

Opcode	Mnemonic	Bytes	Timing	Opcode	Mnemonic	Bytes	Timing
$ED50	in d,(c)	2	12	$ED67	rrd	2	18
$ED51	out (c),d	2	12	$ED68	in l,(c)	2	12
$ED52	sbc hl,de	2	15	$ED69	out (c),l	2	12
$ED53	ld (xx),de	4	20	$ED6A	adc hl,hl	2	15
$ED56	im 1	2	8	$ED6B	ld hl,(xx)	4	20
$ED57	ld a,i	2	9	$ED6F	rld	2	18
$ED58	in e,(c)	2	12	$ED70	in f,(c)	2	12
$ED59	out (c),e	2	12	$ED71	out (c),f	2	12
$ED5A	adc hl,de	2	15	$ED72	sbc hl,sp	2	15
$ED5B	ld de,(xx)	4	20	$ED73	ld (xx),sp	4	20
$ED5E	im 2	2	8	$ED78	in a,(c)	2	12
$ED5F	ld a,r	2	9	$ED79	out (c),a	2	12
$ED60	in h,(c)	2	12	$ED7A	adc hl,sp	2	15
$ED61	out (c),h	2	12	$ED7B	ld sp,(xx)	4	20
$ED62	sbc hl,hl	2	15	$ED8A	push xx	4	*
$ED63	ld (xx),hl	4					

Opcode	Mnemonic	Bytes	Timing	Opcode	Mnemonic	Bytes	Timing
$ED90	outinb *	2	16	$EDAA	ind	2	16
$ED91	nextreg r,v *	4	20	$EDAB	outd	2	16
$ED92	nextreg r,a *	3	17	$EDAC	lddx *	2	16
$ED93	pixeldn *	2	8	$EDB0	ldir	2	21/16
$ED94	pixelad *	2	8	$EDB1	cpir	2	21/16
$ED95	setae *	2	8	$EDB2	inir	2	21/16
$ED98	jp (c) *	2	13	$EDB3	otir	2	21/16
$EDA0	ldi	2	16	$EDB4	ldirx *	2	21/16
$EDA1	cpi	2	16	$EDB7	ldpirx *	2	21/16
$EDA2	ini	2	16	$EDB8	lddr	2	21/16
$EDA3	outi	2	16	$EDB9	cpdr	2	21/16
$EDA4	ldix *	2	16	$EDBA	indr	2	21/16
$EDA5	ldws *	2	14	$EDBB	otdr	2	12/16
$EDA8	ldd	2	16	$EDBC	lddrx *	2	21/16
$EDA9	cpd	2	16				

* ZX Spectrum Next extension

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$FD09	add iy,bc	2	15	$FD35	dec (iy+x)	3	23
$FD19	add iy,de	2	15	$FD36	ld (iy+x),x	5	19
$FD21	ld iy,xx	4	14	$FD39	add iy,sp	2	15
$FD22	ld (xx),iy	4	20	$FD44	ld b,iyh	2	8
$FD23	inc iy	2	10	$FD45	ld b,iyl	2	8
$FD24	inc iyh	2	8	$FD46	ld b,(iy+x)	2	19
$FD25	dec iyh	2	8	$FD4C	ld c,iyh	2	8
$FD26	ld iyh,x	3	11	$FD4D	ld c,iyl	2	8
$FD29	add iy,iy	2	15	$FD4E	ld c,(iy+x)	3	19
$FD2A	ld iy,(xx)	4	20	$FD54	ld d,iyh	2	8
$FD2B	dec iy	2	10	$FD55	ld d,iyl	2	8
$FD2C	inc iyl	2	8	$FD56	ld d,(iy+x)	3	19
$FD2D	dec iyl	2	8	$FD5C	ld e,iyh	2	8
$FD2E	ld iyl,x	4	11	$FD5D	ld e,iyl	2	8
$FD34	inc (iy+x)	3	23	$FD5E	ld e,(iy+x)	3	19

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$FD60	ld iyh,b	2	8	$FD70	ld (iy+x),b	3	19
$FD61	ld iyh,c	2	8	$FD71	ld (iy+x),c	3	19
$FD62	ld iyh,d	2	8	$FD72	ld (iy+x),d	3	19
$FD63	ld iyh,e	2	8	$FD73	ld (iy+x),e	3	19
$FD64	ld iyh,iyh	2	8	$FD74	ld (iy+x),h	3	19
$FD65	ld h,(iy+x)	3	19	$FD75	ld (iy+x),l	3	19
$FD65	ld iyh,iyl	2	8	$FD77	ld (iy+x),a	3	19
$FD67	ld iyh,a	2	8	$FD7C	ld a,iyh	2	8
$FD68	ld iyl,b	2	8	$FD7D	ld a,iyl	2	8
$FD69	ld iyl,c	2	8	$FD7E	ld a,(iy+x)	3	19
$FD6A	ld iyl,d	2	8	$FD84	add a,iyh	2	8
$FD6B	ld iyl,e	2	8	$FD85	add a,iyl	2	8
$FD6C	ld iyl,iyh	2	2	$FD86	add a,(iy+x)	3	19
$FD6D	ld iyl,iyl	2	2	$FD8C	adc a,iyh	2	8
$FD6E	ld l,(iy+x)	3	19	$FD8D	adc a,iyl	2	8
$FD6F	ld iyl,a	2	8	$FD8E	adc a,(iy+x)	3	19

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$FD94	sub iyh	2	8	$FDB4	or iyh	2	8
$FD95	sub iyl	2	8	$FDB5	or iyl	2	8
$FD96	sub (iy+x)	3	19	$FDB6	or (iy+x)	3	19
$FD9C	sbc a,iyh	2	8	$FDBC	cp iyh	2	8
$FD9D	sbc a,iyl	2	8	$FDBD	cp iyl	2	8
$FD9E	sbc a,(iy+x)	3	1	$FDBE	cp (iy+x)	2	19
$FDA4	and iyh	2	8	$FDCB	xBIT+IYx	+1	–
$FDA5	and iyl	2	8	$FDE1	pop iy	2	14
$FDA6	and (iy+x)	3	19	$FDE3	ex (sp),iy	2	23
$FDAC	xor iyh	2	8	$FDE5	push iy	2	15
$FDAD	xor iyl	2	8	$FDE9	jp (iy)	3	8
$FDAE	xor (iy+x)	3	19	$FDF9	ld sp,iy	2	10

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$DDCB06	rlc (ix+x)	4	23	$DDCB0E	rrc (ix+x)	4	23
$DDCB16	rl (ix+x)	4	23	$DDCB1E	rr (ix+x)	4	23
$DDCB26	sla (ix+x)	4	23	$DDCB2E	sra (ix+x)	4	23
$DDCB36	sll (ix+x)	4	23	$DDCB3E	srl (ix+x)	4	23
$DDCB46	bit 0,(ix+x)	4	20	$DDCB4E	bit 1,(ix+x)	4	20
$DDCB56	bit 2,(ix+x)	4	20	$DDCB5E	bit 3,(ix+x)	4	20
$DDCB66	bit 4,(ix+x)	4	20	$DDCB6E	bit 5,(ix+x)	4	20
$DDCB76	bit 6,(ix+x)	4	20	$DDCB7E	bit 7,(ix+x)	4	20
$DDCB86	res 0,(ix+x)	4	23	$DDCB8E	res 1,(ix+x)	4	23
$DDCB96	res 2,(ix+x)	4	23	$DDCB9E	res 3,(ix+x)	4	23
$DDCBA6	res 4,(ix+x)	4	23	$DDCBAE	res 5,(ix+x)	4	23
$DDCBB6	res 6,(ix+x)	4	23	$DDCBBE	res 7,(ix+x)	4	23
$DDCBC6	set 0,(ix+x)	4	23	$DDCBCE	set 1,(ix+x)	4	23
$DDCBD6	set 2,(ix+x)	4	23	$DDCBDE	set 3,(ix+x)	4	23
$DDCBE6	set 4,(ix+x)	4	23	$DDCBEE	set 5,(ix+x)	4	23
$DDCBF6	set 6,(ix+x)	4	23	$DDCBFE	set 7,(ix+x)	4	23

Opcode	Mnemonic	Sz	T	Opcode	Mnemonic	Sz	T
$FDCB06	rlc (iy+x)	4	23	$FDCB0E	rrc (iy+x)	4	23
$FDCB16	rl (iy+x)	4	23	$FDCB1E	rr (iy+x)	4	23
$FDCB26	sla (iy+x)	4	23	$FDCB2E	sra (iy+x)	4	23
$FDCB36	sll (iy+x)	4	23	$FDCB3E	srl (iy+x)	4	23
$FDCB46	bit 0,(iy+x)	4	20	$FDCB4E	bit 1,(iy+x)	4	20
$FDCB56	bit 2,(iy+x)	4	20	$FDCB5E	bit 3,(iy+x)	4	20
$FDCB66	bit 4,(iy+x)	4	20	$FDCB6E	bit 5,(iy+x)	4	20
$FDCB76	bit 6,(iy+x)	4	20	$FDCB7E	bit 7,(iy+x)	4	20
$FDCB86	res 0,(iy+x)	4	23	$FDCB8E	res 1,(iy+x)	4	23
$FDCB96	res 2,(iy+x)	4	23	$FDCB9E	res 3,(iy+x)	4	23
$FDCBA6	res 4,(iy+x)	4	23	$FDCBAE	res 5,(iy+x)	4	23
$FDCBB6	res 6,(iy+x)	4	23	$FDCBBE	res 7,(iy+x)	4	23
$FDCBC6	set 0,(iy+x)	4	23	$FDCBCE	set 1,(iy+x)	4	23
$FDCBD6	set 2,(iy+x)	4	23	$FDCBDE	set 3,(iy+x)	4	23
$FDCBE6	set 4,(iy+x)	4	23	$FDCBEE	set 5,(iy+x)	4	23
$FDCBF6	set 6,(iy+x)	4	23	$FDCBFE	set 7,(iy+x)	4	23