

Z80 timings on Amstrad CPC - Cheat sheet

This document is a visual layout made by cpcitor/findyway from data at <http://www.cpctech.org.uk/docs/instrtim.html>

Instruction timings

The main clock in the CPC is 16Mhz This is provided to the Gate-Array which generates the other clocks.

The Gate Array has the following roles:

generation of a 1Mhz clock for the CRTC and AY-3-8912

generation of a 4Mhz clock for the CPU

arbitrates access to the RAM between the CPU and the video hardware (CRTC and Gate-Array)

Every microsecond:

The CRTC generates a memory address using it's MA and RA signal outputs

The Gate-Array fetches two bytes for each address

The video hardware is given priority so that the display is not disrupted

The Gate-Array generates the "READY" signal which is connected to the "/WAIT" input signal of the CPU. This signal is used to stop the CPU accessing while the video-hardware is accessing it. As a result, all instruction timings are stretched so that they are all multiples of a microsecond (1µs), and this gives an effective CPU clock of 3.3Mhz.

Key:	
cc	condition code (z,nz,c,nc,p,m,po,pe)
r	8-bit register (B,C,D,E,H,L,A)
b	Bit number (0,1,2,3,4,5,6,7)
n	8 bit value
nnnn	16 bit value
dd	8 bit displacement
rp	16-bit register (HL,DE,BC) or SP (except for PUSH and POP)
nc	condition not satisfied
c	condition satisfied

Other timings

Time between acknowledge of a interrupt and execution of a interrupt

Mode 0: (depends on instruction)

Mode 1: 5

Mode 2: 19

1 monitor scanline: 64 microseconds

1 50Hz monitor frame: 19968 microseconds.

NOTES:

(note 1) This timing applies when there are multiple DD or FD prefix's together.

The timings for IY index register pair are identical to the timings for IX register pair.

The table on next page gives the complete execution time for all CPU instructions

These timings have been measured

1 NOP	2 NOPs			3 NOPs		4 NOPs		5 NOPs		6 NOPs		7 NOPs
ARITHMETIC & LOGIC												
ADD A,r ADC A,r SUB r SBC A,r AND r XOR r OR r CP r RLCA RRCA RLA RRA	ADD A,(HL) ADC A,(HL) SUB A,(HL) SBC A,(HL) AND (HL) XOR (HL) OR (HL) CP (HL) RLC r RRC r RR r RL r SLA r SLL r SRL r	ADD A,HIX ADC A,HIX SUB HIX SBC A,HIX AND HIX XOR HIX OR HIX CP HIX	ADD A,n ADC a,n SUB n SBC A,n AND n XOR n OR n CP n	ADD HL,rp		ADD IX,rp	SBC HL,rp ADC HL,rp	ADD A,(IX+dd) ADC A,(IX+dd) SUB (IX+dd) SBC A,(IX+dd) AND (IX+dd) XOR (IX+dd) OR (IX+dd) CP (IX+dd) RLD RRD	CPD CPI			RLC (IX+dd) RRC (IX+dd) RL (IX+dd) RR (IX+dd) SLA (IX+dd) SRA (IX+dd) SLL (IX+dd) SRL (IX+dd)
BITS, FLAGS & SPECIAL ARITHMETIC												
SCF CCF CPL DAA	BIT b,r RES b,r SET b,r NEG			BIT b,(HL)		RES b,(HL) SET b,(HL)					BIT r,(IX+dd)	RES r,(IX+dd) SET r,(IX+dd)
INCR & DECR												
INC r DEC r	INC rp DEC rp	INC HIX DEX HIX	INC LIX DEC LIX	INC (HL) DEC (HL)	INC IX DEC IX						INC (IX+dd) DEC (IX+dd)	
LOAD												
LD r,r	LD A,(BC) LD (BC),A LD (DE),A LD A,(DE) LD r,(HL) LD (HL),r	LD r,HIX LD r,LIX LD HIX,r LD LIX,r	LD r,n	LD (HL),nn	LD HIX,nn LD LIX,nn	LD I,A LD A,I LD R,A LD A,R	LD A,(nnnn) LD (nnnn),A	LD IX,nnnn	LD (nnnn),HL LD HL,(nnnn)	LDI LDD	LD (nnnn),rp LD rp,(nnnn) (includes LD (nnnn),HL and LD (nnnn),HL with ED prefix)	LD (IX+dd),nn LD (nnnn),IX LD IX,(nnnn)
LOAD WITH PC a.k.a. JUMPS												
JP (HL)	JP (IX) JR cc,dd	nc : 2, c : 3		JP nnnn JP cc,nnnn	JR dd RET		DJNZ dd RET RETI	b-1=0 : 3, b-1<>0 : 4				
PUSH/POP/LOAD WITH SP/CALL/RET												
	LD SP,HL			POP rp LD SP,IX			PUSH rp	CALL nnnn RET cc Nc : 2, c : 4	CALL cc,nnnn Nc : 3, c : 5 PUSH IX PUSH IY	POP IX POP IY	EX (SP),HL	EX (SP),IX
IN/OUT, SPECIAL												
NOP HALT EX AF,AF' EX DE,HL EXX DI,EI DD/FD Prefix (note 1)	ED "nop" (ED 00 - ED 3F) IM 0 IM 1 IM 2		IN A,(nn) OUT (nn),A			IN r,(C) OUT (C),r IN F,(C) OUT (C),0 RST 0 RST 1 RST 2 RST 3	RST 4 RST 5 RST 6 RST 7	OUTI OUTD	IND INI	LDIR, LDDR CPDR CPIR INDR, INIR OTDR, OTIR	BC-1=0 : 5, BC-1<>0 : 6 BC-1=0 : 5, BC-1<>0 : 6 BC-1=0 : 5, BC-1<>0 : 6 BC-1=0 : 5, BC-1<>0 : 6	