


Fig. 1.4 Relation between instruction cycle, machine cycle and T-state

Machine Cycle	Status			Control		
	IO/ $\overline{M}$	$S_1$	$S_0$	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read	1	1	0	0	1	1
I/O Write	1	0	1	1	0	1
INTR Acknowledge	1	1	1	1	1	0
Bus Idle	0	0	0	1	1	1

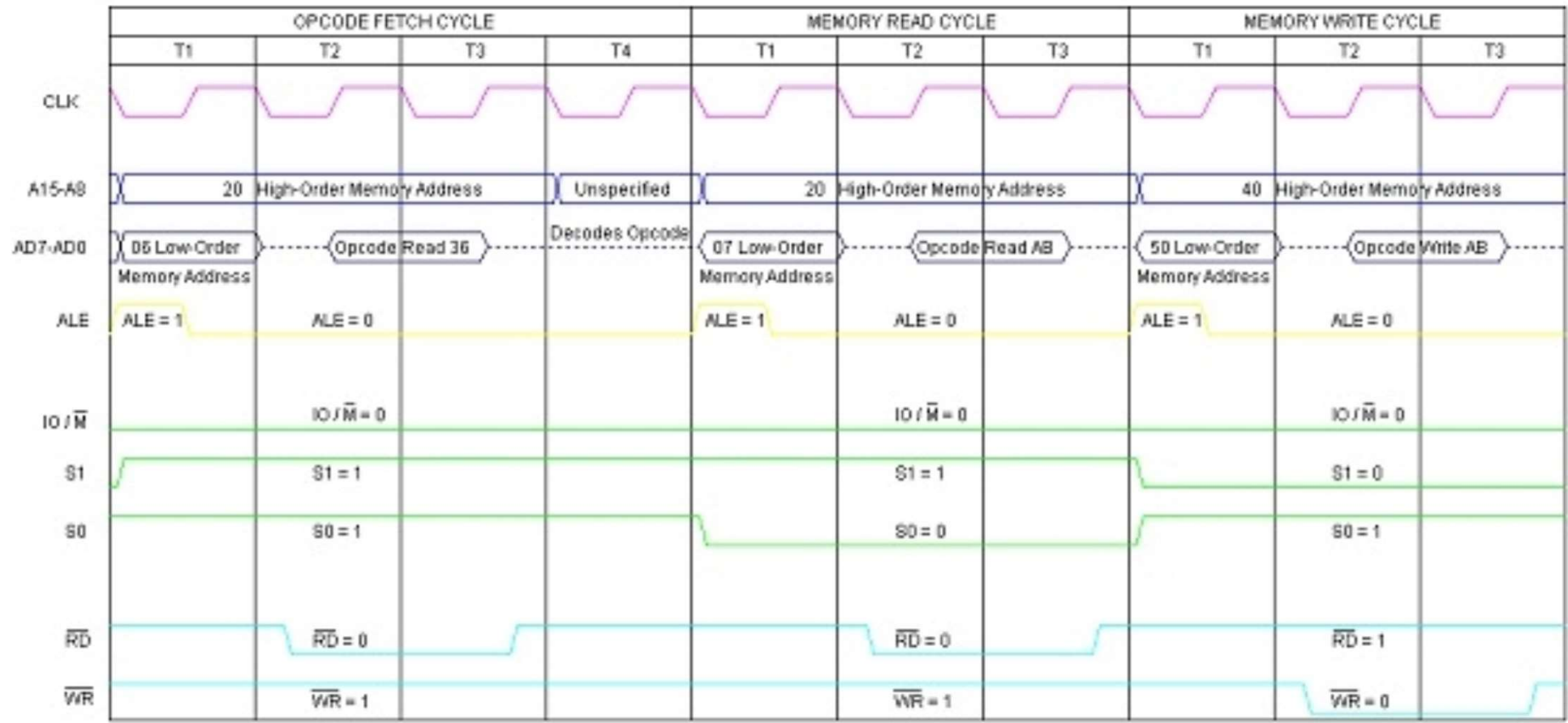
Table 1.1 8085 machine cycles

The fetching, decoding and execution of a single instruction constitutes an **instruction cycle**, which consists of one to five read or write operations between processor and memory or input/output devices.

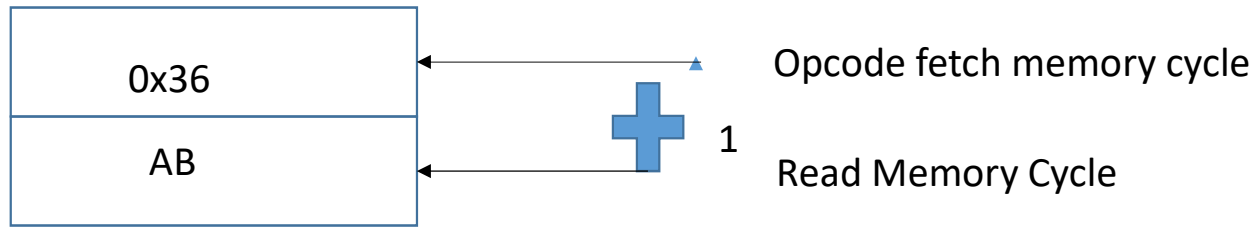
Each memory or I/O operation requires a particular time period, called **machine cycle**. In other words, to move byte of data in or out of the [microprocessor](#), a machine cycle is required. Each machine cycle consists of 3 to 6 clock periods/cycles, referred to as **T-states**. One Instruction Cycle of 8085 consists of one to five machine cycles and one machine cycle consists of three to six T-states i.e. three to six clock periods, as shown in the Fig. 1.4.

MVI M [ 0xAB ] (also written as ABH)   
 Opcode 0x36






The contents of the memory location pointed by the HL pair is updated by the operand AB specified in the program



The part of the program in the memory looks like this:



Finally one WR cycle

1. **Data Transfer Group**  MOV, MVI, LXI, LDA, STA  
MOV r1, ← r2    MOV r, m  
MVI M, data    LDA addr
2. **Arithmetic Group**  ADD, SUB, INR, DAD
3. **Logical Group**  AND, OR, compare, rotate etc.  
Examples are: ANA, XRA, ORA, CMP, and RAL
4. **Branch Control Group**  JMP, JC, JZ, CALL, CZ, RST
5. **I/O and Machine Control Group**  IN, OUT, PUSH, POP, and HLT

1. Write an assembly language program to generate Fibonacci series

<http://lolexfrancis.wordpress.com/2010/03/26/8085-programming-part-1-fibonacci-series/>

Memory	Opcode	Label	Mnemonic	Comments
2000	21 00 25		LXI H,2500	Copy address to HL pair
2003	56		MOV D,M	Transferring the counter to D
2004	3E 00		MVI A,00H	Initialize variable to store sum/next in sequence
2006	06 01		MVI B,01H	Initialize variable to store addend
2008	21 01 25		LXI H,2501	Passing address to H
200B	77	BACK	MOV M,A	Passing data from A to memory
200C	23		INX H	Incrementing memory position
200D	4F		MOV C,A	Passing value from A to C
200E	80		ADD B	A+B and store it in A
200F	41		MOV B,C	Pass value from C to B
2010	15		DCR D	Decrement count
2011	C2 0B 20		JNZ BACK	if D is not 0 go to BACK
2014	76		HLT	Stop

2. Write an assembly language program to find the factorial of a number

<http://lolexfrancis.wordpress.com/2010/03/27/8085-programming-part-2-factorial/>

Memory	Opcode	LABEL	Mnemonic	Comments
2000	3A 00 25		LDA 2500	Loading the value in memory to A
2003	47		MOV B,A	Passing data from A to B
2004	48		MOV C,B	Passing data from B to C
2005	0D		DCR C	Decrement C
2006	59		MOV E,C	Passing data from C to E
2007	1D	LABEL 1	DCR E	Decrement E
2008	CA 17 20		JZ EXIT	If E=0 jump to EXIT
200B	51		MOV D,C	Passing data from C to D
200C	15		DCR D	Decrement D
200D	80	LABEL 2	ADD B	A+B and store it in A
200E	15		DCR D	Decrement D
200F	C2 0D 20		JNZ LABEL 2	If D!=0 go to LABEL2
2012	0D		DCR C	Decrement C
2013	47		MOV B,A	Storing data in A to memory location
2014	C3 07 20		JMP LABEL 1	Go to LABEL1
2017	32 01 25	EXIT	STA 2501	Store content in A to 2501
201A	76		HLT	Stop

**Write an assembly language program to sort 'n'-8 bit numbers in descending order.**

<http://lolexfrancis.wordpress.com/2010/03/29/8085-programming-part-4-sort-into-descending-order/>

<b>Memory</b>	<b>Opcode</b>	<b>Label</b>	<b>Mnemonic</b>	<b>Comments</b>
2000	3A 00 25	DESC	LDA 2500	Load data from memory to A
2003	3D		DCR A	Decrement A
2004	CA 29 20		JZ EXIT	If A equals 0 go to EXIT
2007	21 00 25		LXI H,2500	Load address to HL pair
200A	11 01 25		LXI D,2501	Load address to DE pair
200D	32 00 25		STA 2500	Storing data from A to memory
2010	4F		MOV C,A	Move from A to C
2011	0C		INR C	Increment C
2012	0C		INR C	Increment C
2013	0D	LABEL	DCR C	Decrement C
2014	CA 00 20		JZ DESC	If C equals 0 jump to DESC
2017	23		INX H	Increment HL pair