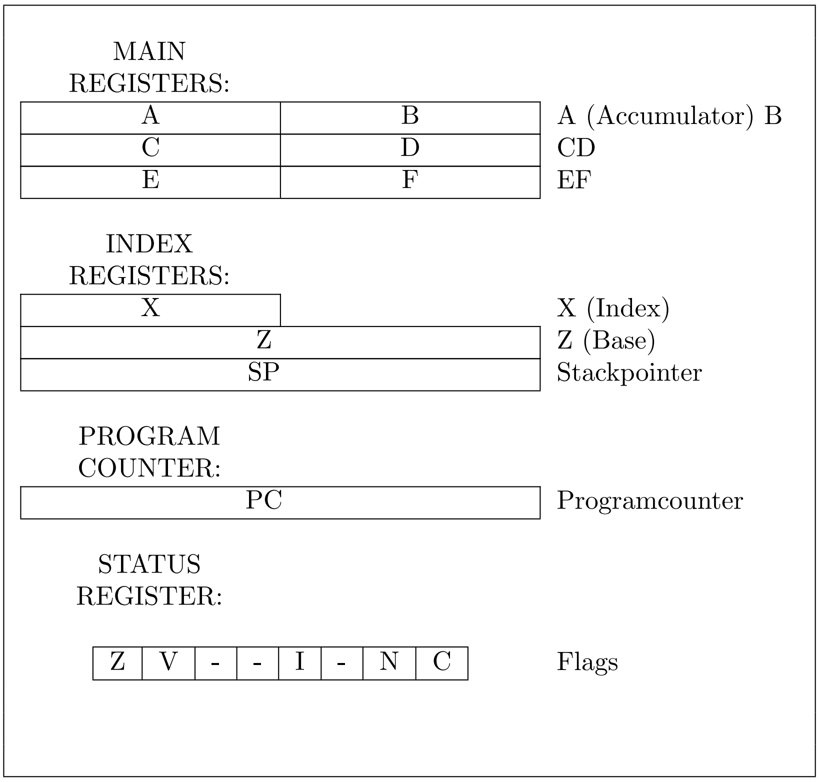
Chimera-2018-A

Assembly Language Programming

Cans Technologies, Inc

Processor Architecture



IMMEDIATE ADDRESSING (#)

The operand is the second byte for 8 bit instructions or the second byte for the lower byte and third byte for the higher byte represent the data for given instruction, no memory addressing is required.

IMPLIED ADDRESSING(impl)

A single byte instruction in which all of the data and operands are implied through the instruction itself.

ABSOLUTE ADDRESSING(abs)

In absolute addressing the second byte of an instruction represents the low order byte of an effective address. The third byte represents the high order byte of an effective address. The two bytes are added to allow full access to 65K of memory.

INDEXED ABSOLUTE ADDRESSING(abs,X)

In indexed absolute addressing the second byte and third byte of an instruction are used in conjunction with a index register (Register X). the second byte of the instruction represents the low order byte of an effective address. The third byte represents the high high byte of an effective address. The result is added to the index register giving a result anywhere in memory. Any 16 bit carry is discarded.

ZERO PAGE ADDRESSING(zpg)

In zero page addressing the second byte of a instruction represents the low order byte of an effective address. The high order byte is fixed at 0 giving you access to the first 256 memory locations.

INDIRECT ADDRESSING(ind)

In indirect addressing the second byte of an instruction represents the low order byte of a full effective address. The third byte represents the high order byte of an effective address forming a full effective address. The contents of the effective address represent the low order byte of an effective address, the contents of the next location in memory represents the high order byte giving the full effective addressing.

BASE OFFSET ADDRESSING(bas)

In base offset addressing, the second instruction byte of the instruction is used as a offset in conjunction with the base register. The offset is calculated by using the given byte as signed, resulting in -128 to +127. This offset is added to the contents of the base register giving the effective address within -128 to +127 of the base register.

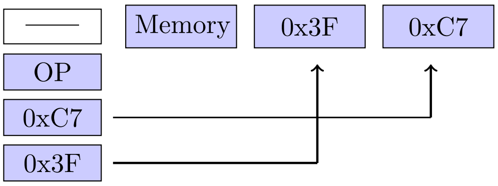
OFFSET ADDRESSING(rel)

In offset addressing, the second instruction byte of the instruction is used as a offset in conjunction with the programcounter. The offset is calculated by using the given byte as signed, resulting in -128 to +127. This offset is added to the contents of the programcounter giving the effective address within -128 to +127.

REGISTER ADDRESSING

In Register addressing the name of the desintation register (and the source where applicable) is stated in the instruction needing no addition bytes.

Little Endian: Any instruction that contains 2 addition byte
are arranged in the order of low first then high.
Below is a example of a opcode using absolute addressing.



Hexadecimal Matrix

-	-	LOW NIBBLE															
-	-	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
HIGH NIBBLE	0x0 -	- -	- -	- -	- -	- -	JP abs	- -	- -	- -	LDX #	LDX abs	LDX abs,X	LDX zpg	LDX (ind)	LDX bas	RTN impl
	0x1 -	ADD A , B	ADD A , C	ADD A , D	ADD A , E	ADD A , F	JCC abs	- -	- -	- -	LODS #	LODS abs	LODS abs,X	LODS zpg	LODS (ind)	LODS bas	RCC impl
	0x2 -	SBB A , B	SBB A , C	SBB A , D	SBB A , E	SBB A , F	JCS abs	TST abs	TST abs,X	TSTA A	LDZ #	LDZ abs	LDZ abs,X	LDZ zpg	LDZ (ind)	LDZ bas	RCS impl
	0x3 -	CMP A , B	CMP A , C	CMP A , D	CMP A , E	CMP A , F	JNE abs	INC abs	INC abs,X	INCA A	- -	STA abs	STA abs,X	STA zpg	STA (ind)	STA bas	RNE impl
	0x4 -	OR A , B	OR A , C	OR A , D	OR A , E	OR A , F	JEQ abs	DEC abs	DEC abs,X	DECA A	- -	- -	- -	- -	- -	- -	REQ impl
	0x5 -	AND A , B	AND A , C	AND A , D	AND A , E	AND A , F	JVC abs	RR abs	RR abs,X	RRA A	MV A , A	MV A , B	MV A , C	MV A , D	MV A , E	MV A , F	RVC impl
	0x6 -	EOR A , B	EOR A , C	EOR A , D	EOR A , E	EOR A , F	JVS abs	RLC abs	RLC abs,X	RLCA A	MV B , A	MV B , B	MV B , C	MV B , D	MV B , E	MV B , F	RVS impl
	0x7 -	STX abs	STX abs,X	STX zpg	STX (ind)	STX bas	JMI abs	ASL abs	ASL abs,X	ASLA A	MV C , A	MV C , B	MV C , C	MV C , D	MV C , E	MV C , F	RMI impl
	0x8 -	- -	ADCP A , C	SBCP A , C	ADI #	- -	JPL abs	ASR abs	ASR abs,X	ASRA A	MV D , A	MV D , B	MV D , C	MV D , D	MV D , E	MV D , F	RPL impl
	0x9 -	DEX impl	XCHG A , C	- -	SBI #	LD B , #	JLS abs	LSR abs	LSR abs,X	LSRA A	MV E , A	MV E , B	MV E , C	MV E , D	MV E , E	MV E , F	RHI impl
	0xa -	INX impl	CLC impl	- -	CPI #	LD C , #	JLT abs	NOT abs	NOT abs,X	NOTA A	MV F , A	MV F , B	MV F , C	MV F , D	MV F , E	MV F , F	RLE impl
	0xb -	DEZ impl	SEC impl	- -	ORI #	LD D , #	MAX impl	NEG abs	NEG abs,X	NEGA A	STS abs	STS abs,X	STS zpg	STS (ind)	STS bas	- -	- -
	0xc -	INZ impl	CLI impl	- -	ANI #	LD E , #	MXA impl	ROL abs	ROL abs,X	ROLA A	STZ abs	STZ abs,X	STZ zpg	STZ (ind)	STZ bas	- -	- -
	0xd -	- -	STI impl	- -	XRI #	LD F , #	CSA impl	RAR abs	RAR abs,X	RARA A	LDA #	LDA abs	LDA abs,X	LDA zpg	LDA (ind)	LDA bas	- -
	0xe -	- -	SEV impl	NOP impl	- -	- -	PSH A	PSH FL	PSH B	PSH C	PSH D	PSH E	PSH F	SWI impl	- -	- -	- -
	0xf -	- -	CLV impl	HLT impl	CALL abs	- -	POP A	POP FL	POP B	POP C	POP D	POP E	POP F	RTI impl	- -	- -	- -

OPCODE		Addressing	Opcode
DESCRIPTION			
Flags:	Z V - - I - N C		
NOTES			

LDA		Addressing	Opcode
Loads Memory into Accumulator		#	0xD9
Flags:	T - - - - - T 0	abs	0xDA
notes		abs,X	0xDB
		zpg	0xDC
		(ind)	0xDD
		bas	0xDE

STA		Addressing	Opcode
Stores Accumulator into Memory		abs	0x3A
Flags:	T - - - - - T 0	abs,X	0x3B
notes		zpg	0x3C
		(ind)	0x3D
		bas	0x3E

ADD		Addressing	Opcode
Register added to Accumulator with Carry		A-B	0x10
Flags:	T T - - - - T T	A-C	0x11
notes		A-D	0x12
		A-E	0x13
		A-F	0x14

SBB		Addressing	Opcode
Register subtracted to Accumulator with Carry		A-B	0x20
Flags:	T T - - - - T T	A-C	0x21
notes		A-D	0x22
		A-E	0x23
		A-F	0x24

CMP		Addressing	Opcode
Register compared to Accumulator		A-B	0x30
Flags:	T T - - - - T T	A-C	0x31
notes		A-D	0x32
		A-E	0x33
		A-F	0x34

OR		Addressing	Opcode
Register bitwise inclusive or with Accumulator		A-B	0x40
Flags:	T - - - - - T -	A-C	0x41
notes		A-D	0x42
		A-E	0x43
		A-F	0x44

AND		Addressing	Opcode
Register bitwise and with Accumulator		A-B	0x50
Flags:	T - - - - - T -	A-C	0x51
notes		A-D	0x52
		A-E	0x53
		A-F	0x54

EOR		Addressing	Opcode
Register bitwise exclusive or with Accumulator		A-B	0x60
Flags:	T - - - - - T -	A-C	0x61
notes		A-D	0x62
		A-E	0x63
		A-F	0x64

ADI		Addressing	Opcode
Data added to Accumulator with Carry		#	0x83
Flags:	T T - - - - T T		
notes			

SBI		Addressing	Opcode
Data subtracted to Accumulator with Carry		#	0x93
Flags:	T T - - - - T T		
notes			

CPI		Addressing	Opcode
Data compared to Accumulator		#	0xA3
Flags:	T T - - - - T T		
notes			

ORI		Addressing	Opcode
Data bitwise inclusive or with Accumulator		#	0xB3
Flags:	T - - - - - T -		
notes			

ANI		Addressing	Opcode
Data bitwise and with Accumulator		#	0xC3
Flags:	T - - - - - T -		
notes			

XRI		Addressing	Opcode
Data bitwise exclusive or with Accumulator		#	0xD3
Flags:	T - - - - - T -		
notes			

TST		Addressing	Opcode
Bit test Memory or Accumulator		abs	0x26
Flags:	T - - - - - T -	abs,X	0x27
notes			

TSTA		Addressing	Opcode
Bit test Memory or Accumulator		A	0x28
Flags:	T - - - - - T -		
notes			

INC		Addressing	Opcode
Increment Memory or Accumulator		abs abs,X	0x36 0x37
Flags:	T - - - - - T -		
notes			

INCA		Addressing	Opcode
Increment Memory or Accumulator		A	0x38
Flags:	T - - - - - T -		
notes			

DEC		Addressing	Opcode
Decrement Memory or Accumulator		abs abs,X	0x46 0x47
Flags:	T - - - - - T -		
notes			

DECA		Addressing	Opcode
Decrement Memory or Accumulator		A	0x48
Flags:	T - - - - - T -		
notes			

RR		Addressing	Opcode
Rotate right through carry Memory or Accumulator		abs abs,X	0x56 0x57
Flags:	T - - - - - T T		
notes			

RRA		Addressing	Opcode
Rotate right through carry Memory or Accumulator		A	0x58
Flags:	T - - - - - T T		
notes			

RLC		Addressing	Opcode
Rotate left through carry Memory or Accumulator		abs abs,X	0x66 0x67
Flags:	T - - - - - T T		
notes			

RLCA		Addressing	Opcode
Rotate left through carry Memory or Accumulator		A	0x68
Flags:	T - - - - - T T		
notes			

ASL		Addressing	Opcode
Arithmetic shift left Memory or Accumulator		abs abs,X	0x76 0x77
Flags:	T - - - - - T T		
notes			

ASLA		Addressing	Opcode
Arithmetic shift left Memory or Accumulator		A	0x78
Flags:	T - - - - - T T		
notes			

ASR		Addressing	Opcode
Arithmetic shift right Memory or Accumulator		abs abs,X	0x86 0x87
Flags:	T - - - - - T T		
notes			

ASRA		Addressing	Opcode
Arithmetic shift right Memory or Accumulator		A	0x88
Flags:	T - - - - - T T		
notes			

LSR		Addressing	Opcode
Shift right Memory or Accumulator		abs abs,X	0x96 0x97
Flags:	T - - - - - T T		
notes			

LSRA		Addressing	Opcode
Shift right Memory or Accumulator		A	0x98
Flags:	T - - - - - T T		
notes			

NOT		Addressing	Opcode
Negate Memory or Accumulator		abs abs,X	0xA6 0xA7
Flags:	T - - - - - T T		
notes			

NOTA		Addressing	Opcode
Negate Memory or Accumulator		A	0xA8
Flags:	T - - - - - T T		
notes			

NEG		Addressing	Opcode
2's complement Memory or Accumulator		abs abs,X	0xB6 0xB7
Flags:	T - - - - - T -		
notes			

NEGA		Addressing	Opcode
2's complement Memory or Accumulator		A	0xB8
Flags:	T - - - - - T -		
notes			

ROL		Addressing	Opcode
Rotate left without carry Memory or Accumulator		abs	0xC6
		abs,X	0xC7
Flags:	T - - - - - T -		
notes			

ROLA		Addressing	Opcode
Rotate left without carry Memory or Accumulator		A	0xC8
Flags:	T - - - - - T -		
notes			

RAR		Addressing	Opcode
Rotate right without carry Memory or Accumulator		abs	0xD6
		abs,X	0xD7
Flags:	T - - - - - T -		
notes			

RARA		Addressing	Opcode
Rotate right without carry Memory or Accumulator		A	0xD8
Flags:	T - - - - - T -		
notes			

MV		Addressing	Opcode
Transfer from one register to another		A-A	0x59
		A-B	0x5A
Flags:	- - - - - - - -	A-C	0x5B
notes		A-D	0x5C
		A-E	0x5D
		A-F	0x5E
		B-A	0x69
		B-B	0x6A
		B-C	0x6B
		B-D	0x6C
		B-E	0x6D
		B-F	0x6E
		C-A	0x79
		C-B	0x7A
		C-C	0x7B
		C-D	0x7C
		C-E	0x7D
		C-F	0x7E
		D-A	0x89
		D-B	0x8A
		D-C	0x8B
		D-D	0x8C
		D-E	0x8D
		D-F	0x8E
		E-A	0x99
		E-B	0x9A
		E-C	0x9B
		E-D	0x9C
		E-E	0x9D
		E-F	0x9E
		F-A	0xA9
		F-B	0xAA
		F-C	0xAB
		F-D	0xAC
		F-E	0xAD
		F-F	0xAE

MAX		Addressing	Opcode
Transters Accumulator to register X		impl	0xB5
Flags:	- - - - - - T -		
notes			

MXA		Addressing	Opcode
Transters register X to Accumulator		impl	0xC5
Flags:	- - - - - - - -		
notes			

LDX		Addressing	Opcode
Loads Memory into register X		#	0x09
		abs	0x0A
Flags:	T - - - - - T 0	abs,X	0x0B
notes		zpg	0x0C
		(ind)	0x0D
		bas	0x0E

STX		Addressing	Opcode
Stores register X into Memory		abs	0x70
		abs,X	0x71
Flags:	T - - - - - T 0	zpg	0x72
notes		(ind)	0x73
		bas	0x74

DEX		Addressing	Opcode
Decrements register X		impl	0x90
Flags:	T - - - - - - -		
notes			

INX		Addressing	Opcode
Increments register X		impl	0xA0
Flags:	T - - - - - - -		
notes			

LODS		Addressing	Opcode
Loads Memory into Stackpointer		#	0x19
		abs	0x1A
Flags:	T - - - - - T 0	abs,X	0x1B
notes		zpg	0x1C
		(ind)	0x1D
		bas	0x1E

STS		Addressing	Opcode
Stores Stackpointer into Memory		abs	0xB9
		abs,X	0xBA
Flags:	T - - - - - T 0	zpg	0xBB
notes		(ind)	0xBC
		bas	0xBD

CSA		Addressing	Opcode
Transters Status register to Accumulator		impl	0xD5
Flags:	- - - - - - - -		
notes			

PSH		Addressing	Opcode
Pushes Register onto the Stack		A	0xE5
		FL	0xE6
Flags:	- - - - - - - -	B	0xE7
notes		C	0xE8
		D	0xE9
		E	0xEA
		F	0xEB

POP		Addressing	Opcode
Pop the top of the Stack into the Register		A	0xF5
		FL	0xF6
Flags:	- - - - - - - -	B	0xF7
notes		C	0xF8
		D	0xF9
		E	0xFA
		F	0xFB

JP		Addressing	Opcode
Loads Memory into ProgramCounter		abs	0x05
Flags:	- - - - - - - -		
notes			

LD		Addressing	Opcode
Loads Memory into register		B,#	0x94
		C,#	0xA4
Flags:	T - - - - - T 0	D,#	0xB4
notes		E,#	0xC4
		F,#	0xD4

ADCP		Addressing	Opcode
Adds register pair into Accumulator pair		A-C	0x81
Flags:	- - - - - - - -		
notes			

SBCP		Addressing	Opcode
Subtracts register pair into Accumulator pair		A-C	0x82
Flags:	- - - - - - - -		
notes			

XCHG		Addressing	Opcode
Swaps the registers contents		A-C	0x91
Flags:	T - - - - - T 0		
notes			

CALL		Addressing	Opcode
Jump to subroutine		abs	0xF3
Flags:	- - - - - - - -		
notes			

RTN		Addressing	Opcode
Return from subroutine		impl	0x0F
Flags:	- - - - - - - -		
notes			

JCC		Addressing	Opcode
Jump on Carry clear		abs	0x15
Flags:	- - - - - - - -		
notes			

JCS		Addressing	Opcode
Jump on Carry set		abs	0x25
Flags:	- - - - - - - -		
notes			

JEQ		Addressing	Opcode
Jump on result equal to Zero		abs	0x45
Flags:	- - - - - - - -		
notes			

JVS		Addressing	Opcode
Jump on overflow set		abs	0x65
Flags:	- - - - - - - -		
notes			

JPL		Addressing	Opcode
Jump on positive result		abs	0x85
Flags:	- - - - - - - -		
notes			

JLT		Addressing	Opcode
Jump on result greater then zero		abs	0xA5
Flags:	- - - - - - - -		
notes			

RCS		Addressing	Opcode
Return on Carry set		impl	0x2F
Flags:	- - - - - - - -		
notes			

REQ		Addressing	Opcode
Return on result equal to Zero		impl	0x4F
Flags:	- - - - - - - -		
notes			

RVS		Addressing	Opcode
Return on overflow set		impl	0x6F
Flags:	- - - - - - - -		
notes			

RPL		Addressing	Opcode
Return on positive result		impl	0x8F
Flags:	- - - - - - - -		
notes			

RLE		Addressing	Opcode
Return on result higher		impl	0xAF
Flags:	- - - - - - - -		
notes			

JNE		Addressing	Opcode
Jump on result not Zero		abs	0x35
Flags:	- - - - - - - -		
notes			

JVC		Addressing	Opcode
Jump on overflow clear		abs	0x55
Flags:	- - - - - - - -		
notes			

JMI		Addressing	Opcode
Jump on negative result		abs	0x75
Flags:	- - - - - - - -		
notes			

JLS		Addressing	Opcode
Jump on result less then zero		abs	0x95
Flags:	- - - - - - - -		
notes			

RCC		Addressing	Opcode
Return on Carry clear		impl	0x1F
Flags:	- - - - - - - -		
notes			

RNE		Addressing	Opcode
Return on result not Zero		impl	0x3F
Flags:	- - - - - - - -		
notes			

RVC		Addressing	Opcode
Return on overflow clear		impl	0x5F
Flags:	- - - - - - - -		
notes			

RMI		Addressing	Opcode
Return on negative result		impl	0x7F
Flags:	- - - - - - - -		
notes			

RHI		Addressing	Opcode
Return on result same or lower		impl	0x9F
Flags:	- - - - - - - -		
notes			

CLC		Addressing	Opcode
Clear Carry flag		impl	0xA1
Flags:	- - - - - - - 0		
notes			

SEC		Addressing	Opcode
Set Carry flag		impl	0xB1
Flags:	- - - - - - - 1		
notes			

STI		Addressing	Opcode
Set Interupt flag		impl	0xD1
Flags:	- - - - 1 - - -		
notes			

CLV		Addressing	Opcode
Clear Overflow flag		impl	0xF1
Flags:	- 0 - - - - - -		
notes			

HLT		Addressing	Opcode
Wait for interupt		impl	0xF2
Flags:	- - - - - - - -		
notes			

RTI		Addressing	Opcode
Return from software interupt		impl	0xFC
Flags:	- - - - - - - -		
Pops: General purpose registers (in order) Staus register ProgramCounter Accumulator			

STZ		Addressing	Opcode
Stores register Z into Memory		abs	0xC9
		abs,X	0xCA
Flags:	- - - - - - - -	zpg	0xCB
notes		(ind)	0xCC
		bas	0xCD

INZ		Addressing	Opcode
Increments base register		impl	0xC0
Flags:	T - - - - - - -		
notes			

CLI		Addressing	Opcode
Clear Interupt flag		impl	0xC1
Flags:	- - - - 0 - - -		
notes			

SEV		Addressing	Opcode
Set Overflow flag		impl	0xE1
Flags:	- 1 - - - - - -		
notes			

NOP		Addressing	Opcode
No operation		impl	0xE2
Flags:	- - - - - - - -		
notes			

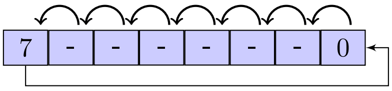
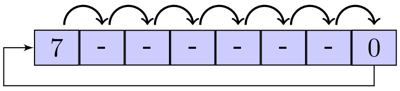
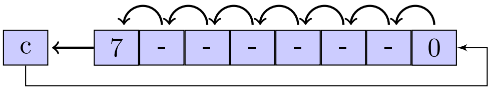
SWI		Addressing	Opcode
Software interupt		impl	0xEC
Flags:	- - - - 1 - - -		
Pushes: Accumulator ProgramCounter Staus register General purpose registers (in order)			

LDZ		Addressing	Opcode
Loads Memory into register Z		#	0x29
		abs	0x2A
Flags:	- - - - - - - -	abs,X	0x2B
notes		zpg	0x2C
		(ind)	0x2D
		bas	0x2E

DEZ		Addressing	Opcode
Decrements base register		impl	0xB0
Flags:	T - - - - - - -		
notes			

			Registers Source								
op	details	Dest	A	FL	B	C	D	E	F	Stack	FLags
ADD	A + CF + R	A	-	-	0x10	0x11	0x12	0x13	0x14	-	T T - - - - T T
SBB	A - CF - R	A	-	-	0x20	0x21	0x22	0x23	0x24	-	T T - - - - T T
CMP	A - R		-	-	0x30	0x31	0x32	0x33	0x34	-	T T - - - - T T
OR	A |R	A	-	-	0x40	0x41	0x42	0x43	0x44	-	T - - - - - T -
AND	A & R	A	-	-	0x50	0x51	0x52	0x53	0x54	-	T - - - - - T -
EOR	A (+) R	A	-	-	0x60	0x61	0x62	0x63	0x64	-	T - - - - - T -
TSTA	A - 0	A	0x28	-	-	-	-	-	-	-	T - - - - - T -
INCA	A + 1	A	0x38	-	-	-	-	-	-	-	T - - - - - T -
DECA	A - 1	A	0x48	-	-	-	-	-	-	-	T - - - - - T -
RRA	fig 7	A	0x58	-	-	-	-	-	-	-	T - - - - - T T
RLCA	fig 6	A	0x68	-	-	-	-	-	-	-	T - - - - - T T
ASLA	fig 1	A	0x78	-	-	-	-	-	-	-	T - - - - - T T
ASRA	fijg 2	A	0x88	-	-	-	-	-	-	-	T - - - - - T T
LSRA	fig 3	A	0x98	-	-	-	-	-	-	-	T - - - - - T T
NOTA	A ~	A	0xA8	-	-	-	-	-	-	-	T - - - - - T T
NEGA	0 - A	A	0xB8	-	-	-	-	-	-	-	T - - - - - T -
ROLA	fig 5	A	0xC8	-	-	-	-	-	-	-	T - - - - - T -
RARA	fig 4	A	0xD8	-	-	-	-	-	-	-	T - - - - - T -
MV	A	A	0x59	-	0x5A	0x5B	0x5C	0x5D	0x5E	-	
		B	0x69	-	0x6A	0x6B	0x6C	0x6D	0x6E	-	
		C	0x79	-	0x7A	0x7B	0x7C	0x7D	0x7E	-	
		D	0x89	-	0x8A	0x8B	0x8C	0x8D	0x8E	-	
		E	0x99	-	0x9A	0x9B	0x9C	0x9D	0x9E	-	
		F	0xA9	-	0xAA	0xAB	0xAC	0xAD	0xAE	-	- - - - - - - -
PSH	A -*		0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	-	- - - - - - - -
POP	+*	A	-	-	-	-	-	-	-	0xF5	
		FL	-	-	-	-	-	-	-	0xF6	
		B	-	-	-	-	-	-	-	0xF7	
		C	-	-	-	-	-	-	-	0xF8	
		D	-	-	-	-	-	-	-	0xF9	
		E	-	-	-	-	-	-	-	0xFA	
		F	-	-	-	-	-	-	-	0xFB	- - - - - - - -
ADCP	R + A	A	-	-	-	0x81	-	-	-	-	- - - - - - - -
SBCP	R - A	A	-	-	-	0x82	-	-	-	-	- - - - - - - -
XCHG	R	A	-	-	-	0x91	-	-	-	-	T - - - - - T 0

op	details	Dest	#	impl	abs	abs,X	zpg	(ind)	bas	rel	FLags
LDA	M	A	0xD9	-	0xDA	0xDB	0xDC	0xDD	0xDE	-	T - - - - - T 0
STA	A	M	-	-	0x3A	0x3B	0x3C	0x3D	0x3E	-	T - - - - - T 0
ADI	A + CF + M	A	0x83	-	-	-	-	-	-	-	T T - - - - T T
SBI	A - CF - M	A	0x93	-	-	-	-	-	-	-	T T - - - - T T
CPI	A - M		0xA3	-	-	-	-	-	-	-	T T - - - - T T
ORI	A |M	A	0xB3	-	-	-	-	-	-	-	T - - - - - T -
ANI	A & M	A	0xC3	-	-	-	-	-	-	-	T - - - - - T -
XRI	A (+) M	A	0xD3	-	-	-	-	-	-	-	T - - - - - T -
TST	M - 0		-	-	0x26	0x27	-	-	-	-	T - - - - - T -
INC	M + 1	M	-	-	0x36	0x37	-	-	-	-	T - - - - - T -
DEC	M - 1	M	-	-	0x46	0x47	-	-	-	-	T - - - - - T -
RR	fig 7	M	-	-	0x56	0x57	-	-	-	-	T - - - - - T T
RLC	fig 6	M	-	-	0x66	0x67	-	-	-	-	T - - - - - T T
ASL	fig 1	M	-	-	0x76	0x77	-	-	-	-	T - - - - - T T
ASR	fijg 2	M	-	-	0x86	0x87	-	-	-	-	T - - - - - T T
LSR	fig 3	M	-	-	0x96	0x97	-	-	-	-	T - - - - - T T
NOT	M ∼	M	-	-	0xA6	0xA7	-	-	-	-	T - - - - - T T
NEG	0 - M	M	-	-	0xB6	0xB7	-	-	-	-	T - - - - - T -
ROL	fig 5	M	-	-	0xC6	0xC7	-	-	-	-	T - - - - - T -
RAR	fig 4	M	-	-	0xD6	0xD7	-	-	-	-	T - - - - - T -
MAX	A	X	-	0xB5	-	-	-	-	-	-	- - - - - - T -
MXA	X	A	-	0xC5	-	-	-	-	-	-	- - - - - - - -
LDX	M	X	0x09	-	0x0A	0x0B	0x0C	0x0D	0x0E	-	T - - - - - T 0
STX	X	M	-	-	0x70	0x71	0x72	0x73	0x74	-	T - - - - - T 0
DEX	X - 1	X	-	0x90	-	-	-	-	-	-	T - - - - - - -
INX	X + 1	X	-	0xA0	-	-	-	-	-	-	T - - - - - - -
LODS	M	SP	0x19	-	0x1A	0x1B	0x1C	0x1D	0x1E	-	T - - - - - T 0
STS	SP	M	-	-	0xB9	0xBA	0xBB	0xBC	0xBD	-	T - - - - - T 0
CSA	FL	A	-	0xD5	-	-	-	-	-	-	- - - - - - - -
JP			-	-	0x05	-	-	-	-	-	- - - - - - - -
LD	M	B	0x94	-	-	-	-	-	-	-	
		C	0xA4	-	-	-	-	-	-	-	
		D	0xB4	-	-	-	-	-	-	-	
		E	0xC4	-	-	-	-	-	-	-	
		F	0xD4	-	-	-	-	-	-	-	T - - - - - T 0
CALL			-	-	0xF3	-	-	-	-	-	- - - - - - - -
RTN			-	0x0F	-	-	-	-	-	-	- - - - - - - -
JCC	CF = 0		-	-	0x15	-	-	-	-	-	- - - - - - - -
JCS	CF = 1		-	-	0x25	-	-	-	-	-	- - - - - - - -
JNE	ZF = 0		-	-	0x35	-	-	-	-	-	- - - - - - - -
JEQ	ZF = 1		-	-	0x45	-	-	-	-	-	- - - - - - - -
JVC	VF = 0		-	-	0x55	-	-	-	-	-	- - - - - - - -
JVS	VF = 1		-	-	0x65	-	-	-	-	-	- - - - - - - -
JMI	NF = 1		-	-	0x75	-	-	-	-	-	- - - - - - - -
JPL	NF = 0		-	-	0x85	-	-	-	-	-	- - - - - - - -
JLS	ZF |NF ˆVF = 0		-	-	0x95	-	-	-	-	-	- - - - - - - -
JLT	NF ˆVF = 1		-	-	0xA5	-	-	-	-	-	- - - - - - - -
RCC	CF = 0		-	0x1F	-	-	-	-	-	-	- - - - - - - -
RCS	CF = 1		-	0x2F	-	-	-	-	-	-	- - - - - - - -
RNE	ZF = 0		-	0x3F	-	-	-	-	-	-	- - - - - - - -
REQ	ZF = 1		-	0x4F	-	-	-	-	-	-	- - - - - - - -
RVC	VF = 0		-	0x5F	-	-	-	-	-	-	- - - - - - - -
RVS	VF = 1		-	0x6F	-	-	-	-	-	-	- - - - - - - -
RMI	NF = 1		-	0x7F	-	-	-	-	-	-	- - - - - - - -
RPL	NF = 0		-	0x8F	-	-	-	-	-	-	- - - - - - - -
RHI	CF |ZF = 1		-	0x9F	-	-	-	-	-	-	- - - - - - - -
RLE	CF |ZF = 0		-	0xAF	-	-	-	-	-	-	- - - - - - - -
CLC	CF = 0		-	0xA1	-	-	-	-	-	-	- - - - - - - 0
SEC	CF = 1		-	0xB1	-	-	-	-	-	-	- - - - - - - 1
CLI	IF = 0		-	0xC1	-	-	-	-	-	-	- - - - 0 - - -
STI	IF = 1		-	0xD1	-	-	-	-	-	-	- - - - 1 - - -
SEV	VF = 1		-	0xE1	-	-	-	-	-	-	- 1 - - - - - -
CLV	VF = 0		-	0xF1	-	-	-	-	-	-	- 0 - - - - - -
NOP			-	0xE2	-	-	-	-	-	-	- - - - - - - -
HLT			-	0xF2	-	-	-	-	-	-	- - - - - - - -
SWI	-		-	0xEC	-	-	-	-	-	-	- - - - 1 - - -
RTI	-		-	0xFC	-	-	-	-	-	-	- - - - - - - -
LDZ	M	Z	0x29	-	0x2A	0x2B	0x2C	0x2D	0x2E	-	- - - - - - - -
STZ	Z	M	-	-	0xC9	0xCA	0xCB	0xCC	0xCD	-	- - - - - - - -
DEZ	Z - 1	Z	-	0xB0	-	-	-	-	-	-	T - - - - - - -
INZ	Z + 1	Z	-	0xC0	-	-	-	-	-	-	T - - - - - - -

Key	
A - Accumulator	SP - StackPointer
FL - Status Register	M - Memory
|- Inclusive or	(+) - Exclusive or
& - logical and	∼ - Negation
-* - Push to stack and decrement stack pointer	+* - Increment stack pointer and pop from stack
R - General Register	X - Index Register
CF - Carry FLag	ZF - Zero FLag
NF - Negative FLag	IF - Zero FLag
VF - OverflowFlag FLag	
Z - Base Register	
FIGURE 1: C $\leftarrow \boxed{7 \qquad\qquad 0} \leftarrow 0$	FIGURE 2: N $\rightarrow \boxed{7 \qquad\qquad 0} \rightarrow C$
FIGURE 3: $0 \rightarrow \boxed{7 \qquad\qquad 0} \rightarrow C$	
FIGURE 4: 	FIGURE 5 
FIGURE 6: 	FIGURE 7