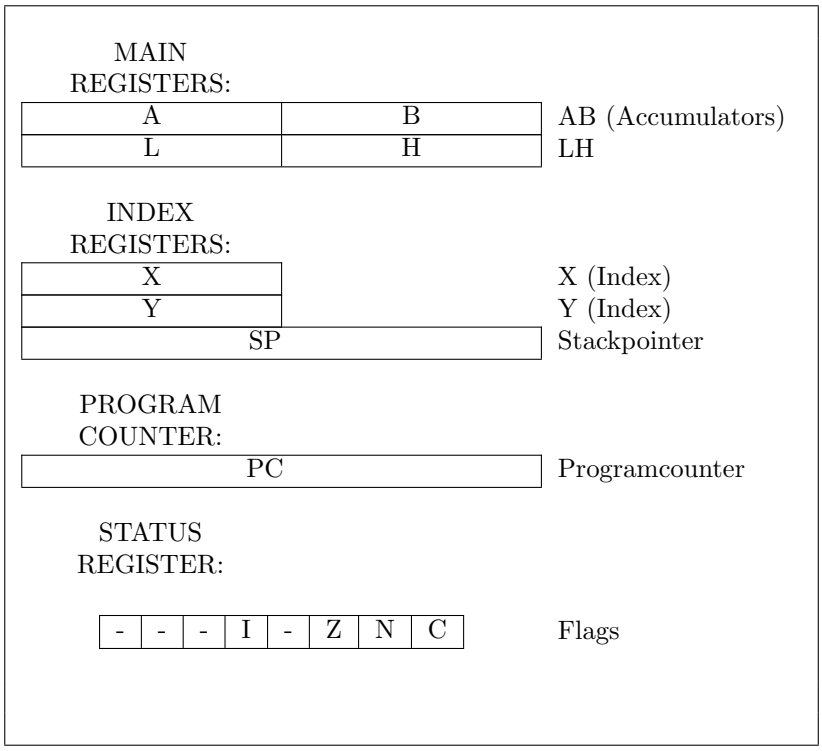
Chimera-2016-A

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 upper byte and third byte for the lower 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 high order byte of an effective address. The third byte represents the low 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 or Register Y). the second byte of the instruction represents the high order byte of an effective address. The third byte represents the high low 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.

INDIRECT ADDRESSING(ind)

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

INDEXED INDIRECT ADDRESSING((ind,X))

In indexed indirect addressing, the second byte (high order) and third byte(low order) of an instruction is added to a index register (Register X or Register Y) a location in memory. The contents of this address represents the high order byte of an effective address, the contents of the next location in memory represents the low order byte giving the full effective addressing.

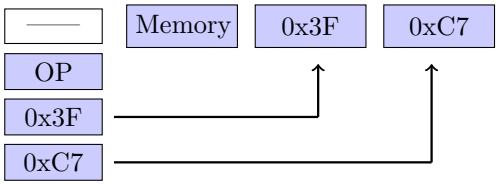
OFFSET ADDRESSING(rel)

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.

Big Endian: Any instruction that contains 2 addition byte are arranged in the order of high first then low.
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	-	DECX	INCX	DEY	INCY	CLC	STC	CLI	STI	-	LDAA	LDAB	LX	LX	LDX	LDY
	-	-	impl	impl	impl	impl	impl	impl	impl	impl	-	#	#	#	#	#	#
	0x1	JMP	JCC	JCS	JNE	JEQ	JMI	JPL	JHI	JLE	-	LDAA	LDAB	MVI	MVI	LDX	LDY
	-	abs	abs	abs	abs	abs	abs	abs	abs	abs	-	abs	abs	L , #	H , #	abs	abs
	0x2	LODS	JSR	CCC	CCS	CNE	CEQ	CMI	CPL	CHI	CLE	LDAA	LDAB	NOP	HLT	LDX	LDY
	-	#	abs	abs	abs	abs	abs	abs	abs	abs	abs	abs,X	abs,X	impl	impl	abs,X	abs,X
	0x3	LODS	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	LDAA	LDAB	-	-	LDX	LDY
	-	abs	A , L	A , L	A , L	A , L	A , L	A , L	A , L	A , L	A , L	abs,Y	abs,Y	-	-	abs,Y	abs,Y
	0x4	LODS	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	LDAA	LDAB	RET	-	LDX	LDY
	-	abs,X	A , H	A , H	A , H	A , H	A , H	A , H	A , H	A , H	A , H	(ind)	(ind)	impl	-	(ind)	(ind)
	0x5	LODS	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	LDAA	LDAB	SWI	RTI	LDX	LDY
	-	abs,Y	A , M	A , M	A , M	A , M	A , M	A , M	A , M	A , M	A , M	(ind,X)	(ind,X)	impl	impl	(ind,X)	(ind,X)
	0x6	LODS	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	STOS	MOVE	MOVE	MOVE	MOVE	MOVE
	-	(ind)	B , L	B , L	B , L	B , L	B , L	B , L	B , L	B , L	B , L	abs	A , A	B , A	L , A	H , A	M , A
	0x7	LODS	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	STOS	MOVE	MOVE	MOVE	MOVE	MOVE
	-	(ind,X)	B , H	B , H	B , H	B , H	B , H	B , H	B , H	B , H	B , H	abs,X	A , B	B , B	L , B	H , B	M , B
	0x8	-	ADC	SBC	ADD	SUB	CMP	OR	AND	XOR	BIT	STOS	MOVE	MOVE	MOVE	MOVE	MOVE
	-	-	B , M	B , M	B , M	B , M	B , M	B , M	B , M	B , M	B , M	abs,Y	A , L	B , L	L , L	H , L	M , L
	0x9	-	-	-	SBIA	SBIB	CPIA	CPIB	ORIA	ORIB	-	STOS	MOVE	MOVE	MOVE	MOVE	MOVE
	-	-	-	-	#	#	#	#	#	#	-	(ind)	A , H	B , H	L , H	H , H	M , H
	0xa	INC	DEC	RRC	RLC	SAL	SAR	LSR	COM	ROL	RR	STOS	MOVE	MOVE	MOVE	MOVE	MOVE
	-	abs	abs	abs	abs	abs	abs	abs	abs	abs	abs	(ind,X)	A , M	B , M	L , M	H , M	- , -
	0xb	INC	DEC	RRC	RLC	SAL	SAR	LSR	COM	ROL	RR	STORA	STORB	STOX	STOY	PUSH	POP
	-	abs,X	abs,X	abs,X	abs,X	abs,X	abs,X	abs,X	abs,X	abs,X	abs,X	abs	abs	abs	abs	A	A
	0xc	INC	DEC	RRC	RLC	SAL	SAR	LSR	COM	ROL	RR	STORA	STORB	STOX	STOY	PUSH	POP
	-	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,Y	abs,X	abs,X	abs,X	abs,X	B	B
	0xd	INCA	DECA	RRCA	RLCA	SALA	SARA	LSRA	COMA	ROLA	RRA	STORA	STORB	STOX	STOY	PUSH	POP
	-	A	A	A	A	A	A	A	A	A	A	abs,Y	abs,Y	abs,Y	abs,Y	FL	FL
	0xe	INCB	DECB	RRCB	RLCB	SALB	SARB	LSRB	COMB	ROLB	RRB	STORA	STORB	STOX	STOY	PUSH	POP
	-	B	B	B	B	B	B	B	B	B	B	(ind)	(ind)	(ind)	(ind)	L	L
	0xf	CAY	MYA	CSA	ABA	SBA	AAB	SAB	ADCP	SBCP	XCHG	STORA	STORB	STOX	STOY	PUSH	POP
	-	impl	impl	impl	impl	impl	impl	impl	A , L	A , L	A , L	(ind,X)	(ind,X)	(ind,X)	(ind,X)	H	H

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

LDAA		Addressing	Opcode
Loads Memory into Accumulator		#	0x0A
Flags:	- - - - - - - -	abs	0x1A
notes		abs,X	0x2A
		abs,Y	0x3A
		(ind)	0x4A
		(ind,X)	0x5A

LDAB		Addressing	Opcode
Loads Memory into Accumulator		#	0x0B
Flags:	- - - - - - - -	abs	0x1B
notes		abs,X	0x2B
		abs,Y	0x3B
		(ind)	0x4B
		(ind,X)	0x5B

STORA		Addressing	Opcode
Stores Accumulator into Memory		abs	0xBA
Flags:	- - - - - - - -	abs,X	0xCA
notes		abs,Y	0xDA
		(ind)	0xEA
		(ind,X)	0xFA

STORB		Addressing	Opcode
Stores Accumulator into Memory		abs	0xBB
Flags:	- - - - - - - -	abs,X	0xCB
notes		abs,Y	0xDB
		(ind)	0xEB
		(ind,X)	0xFB

ADC		Addressing	Opcode
Register added to Accumulator with Carry		A-L	0x31
Flags:	- - - - - T T T	A-H	0x41
notes		A-M	0x51
		B-L	0x61
		B-H	0x71
		B-M	0x81

SBC		Addressing	Opcode
Register subtracted to Accumulator with Carry		A-L	0x32
Flags:	- - - - - T T T	A-H	0x42
notes		A-M	0x52
		B-L	0x62
		B-H	0x72
		B-M	0x82

ADD		Addressing	Opcode
Register added to Accumulator		A-L	0x33
Flags:	- - - - - T T T	A-H	0x43
notes		A-M	0x53
		B-L	0x63
		B-H	0x73
		B-M	0x83

SUB		Addressing	Opcode
Register subtracted to Accumulator		A-L	0x34
Flags:	- - - - - T T T	A-H	0x44
notes		A-M	0x54
		B-L	0x64
		B-H	0x74
		B-M	0x84

CMP		Addressing	Opcode
Register compared to Accumulator		A-L	0x35
Flags:	- - - - - T T T	A-H	0x45
notes		A-M	0x55
		B-L	0x65
		B-H	0x75
		B-M	0x85

OR		Addressing	Opcode
Register bitwise inclusive or with Accumulator		A-L	0x36
Flags:	- - - - - T T -	A-H	0x46
notes		A-M	0x56
		B-L	0x66
		B-H	0x76
		B-M	0x86

AND	Addressing	Opcode
Register bitwise and with Accumulator	A-L	0x37
	A-H	0x47
Flags: - - - - - T T -	A-M	0x57
	B-L	0x67
notes	B-H	0x77
	B-M	0x87

XOR	Addressing	Opcode
Register bitwise exclusive or with Accumulator	A-L	0x38
	A-H	0x48
	A-M	0x58
Flags: - - - - - T T -	B-L	0x68
	B-H	0x78
notes	B-M	0x88

BIT	Addressing	Opcode
Register Bit tested with Accumulator	A-L	0x39
	A-H	0x49
Flags: - - - - - T T -	A-M	0x59
	B-L	0x69
notes	B-H	0x79
	B-M	0x89

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

SBIB	Addressing	Opcode
Data subtracted to Accumulator with Carry	#	0x94
Flags: - - - - - T T T		
notes		

CPIA	Addressing	Opcode
Data compared to Accumulator	#	0x95
Flags: - - - - - T T T		
notes		

CPIB	Addressing	Opcode
Data compared to Accumulator	#	0x96
Flags: - - - - - T T T		
notes		

ORIA	Addressing	Opcode
Data bitwise inclusive or with Accumulator	#	0x97
Flags: - - - - - T T -		
notes		

ORIB	Addressing	Opcode
Data bitwise inclusive or with Accumulator	#	0x98
Flags: - - - - - T T -		
notes		

INC	Addressing	Opcode
Increment Memory or Accumulator	abs	0xA0
	abs,X	0xB0
Flags: - - - - - T T -	abs,Y	0xC0
notes		

INCA	Addressing	Opcode
Increment Memory or Accumulator	A	0xD0
Flags: - - - - - T T -		
notes		

INCB	Addressing	Opcode
Increment Memory or Accumulator	B	0xE0
Flags: - - - - - T T -		
notes		

DEC	Addressing	Opcode
Decrement Memory or Accumulator	abs	0xA1
	abs,X	0xB1
Flags: - - - - - T T -	abs,Y	0xC1
notes		

DECA	Addressing	Opcode
Decrement Memory or Accumulator	A	0xD1
Flags: - - - - - T T -		
notes		

DECB	Addressing	Opcode
Decrement Memory or Accumulator	B	0xE1
Flags: - - - - - T T -		
notes		

RRC	Addressing	Opcode
Rotate right through carry Memory or Accumulator	abs	0xA2
	abs,X	0xB2
Flags: - - - - - T T T	abs,Y	0xC2
notes		

RRCA	Addressing	Opcode
Rotate right through carry Memory or Accumulator	A	0xD2
Flags: - - - - - T T T		
notes		

RRCB	Addressing	Opcode
Rotate right through carry Memory or Accumulator	B	0xE2
Flags: - - - - - T T T		
notes		

RLC	Addressing	Opcode
Rotate left through carry Memory or Accumulator	abs	0xA3
	abs,X	0xB3
	abs,Y	0xC3
Flags: - - - - - T T T		
notes		

RLCA	Addressing	Opcode
Rotate left through carry Memory or Accumulator	A	0xD3
Flags: - - - - - T T T		
notes		

RLCB	Addressing	Opcode
Rotate left through carry Memory or Accumulator	B	0xE3
Flags: - - - - - T T T		
notes		

SAL	Addressing	Opcode
Arithmetic shift left Memory or Accumulator	abs	0xA4
	abs,X	0xB4
	abs,Y	0xC4
Flags: - - - - - T T T		
notes		

SALA	Addressing	Opcode
Arithmetic shift left Memory or Accumulator	A	0xD4
Flags: - - - - - T T T		
notes		

SALB	Addressing	Opcode
Arithmetic shift left Memory or Accumulator	B	0xE4
Flags: - - - - - T T T		
notes		

SAR	Addressing	Opcode
Arithmetic shift right Memory or Accumulator	abs	0xA5
	abs,X	0xB5
	abs,Y	0xC5
Flags: - - - - - T T T		
notes		

SARA	Addressing	Opcode
Arithmetic shift right Memory or Accumulator	A	0xD5
Flags: - - - - - T T T		
notes		

SARB	Addressing	Opcode
Arithmetic shift right Memory or Accumulator	B	0xE5
Flags: - - - - - T T T		
notes		

LSR	Addressing	Opcode
Shift right Memory or Accumulator	abs	0xA6
	abs,X	0xB6
	abs,Y	0xC6
Flags: - - - - - T T T		
notes		

LSRA	Addressing	Opcode
Shift right Memory or Accumulator	A	0xD6
Flags: - - - - - T T T		
notes		

LSRB	Addressing	Opcode
Shift right Memory or Accumulator	B	0xE6
Flags: - - - - - T T T		
notes		

COM	Addressing	Opcode
Negate Memory or Accumulator	abs	0xA7
	abs,X	0xB7
Flags: - - - - - T T T	abs,Y	0xC7
notes		

COMA	Addressing	Opcode
Negate Memory or Accumulator	A	0xD7
Flags: - - - - - T T T		
notes		

COMB	Addressing	Opcode
Negate Memory or Accumulator	B	0xE7
Flags: - - - - - T T T		
notes		

ROL	Addressing	Opcode
Rotate left without carry Memory or Accumulator	abs	0xA8
	abs,X	0xB8
	abs,Y	0xC8
Flags: - - - - - T T -		
notes		

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

ROLB	Addressing	Opcode
Rotate left without carry Memory or Accumulator	B	0xE8
Flags: - - - - - T T -		
notes		

RR	Addressing	Opcode
Rotate right without carry Memory or Accumulator	abs	0xA9
	abs,X	0xB9
	abs,Y	0xC9
Flags: - - - - - T T -		
notes		

RRA	Addressing	Opcode
Rotate right without carry Memory or Accumulator	A	0xD9
Flags: - - - - - T T -		
notes		

MOVE	Addressing	Opcode
Transfer from one register to another	A-A	0x6B
	A-B	0x7B
Flags: - - - - - - - -	A-L	0x8B
notes	A-H	0x9B
	A-M	0xAB
	B-A	0x6C
	B-B	0x7C
	B-L	0x8C
	B-H	0x9C
	B-M	0xAC
	L-A	0x6D
	L-B	0x7D
	L-L	0x8D
	L-H	0x9D
	L-M	0xAD
	H-A	0x6E
	H-B	0x7E
	H-L	0x8E
	H-H	0x9E
	H-M	0xAE
	M-A	0x6F
	M-B	0x7F
	M-L	0x8F
	M-H	0x9F
	—	0xAF

RRB	Addressing	Opcode
Rotate right without carry Memory or Accumulator	B	0xE9
Flags: - - - - - T T -		
notes		

LDX	Addressing	Opcode
Loads Memory into register X	#	0x0E
	abs	0x1E
Flags: - - - - - - - -	abs,X	0x2E
notes	abs,Y	0x3E
	(ind)	0x4E
	(ind,X)	0x5E

STOX	Addressing	Opcode
Stores register X into Memory	abs	0xBC
	abs,X	0xCC
Flags: - - - - - - - -	abs,Y	0xDC
notes	(ind)	0xEC
	(ind,X)	0xFC

DECX	Addressing	Opcode
Decrements register X	impl	0x01
Flags: - - - - - T - -		
notes		

INCX	Addressing	Opcode
Increments register X	impl	0x02
Flags: - - - - - T - -		
notes		

LDY	Addressing	Opcode
Loads Memory into register Y	#	0x0F
	abs	0x1F
Flags: - - - - - - - -	abs,X	0x2F
notes	abs,Y	0x3F
	(ind)	0x4F
	(ind,X)	0x5F

STOY	Addressing	Opcode
Stores register Y into Memory	abs	0xBD
	abs,X	0xCD
Flags: - - - - - - - -	abs,Y	0xDD
notes	(ind)	0xED
	(ind,X)	0xFD

CAY	Addressing	Opcode
Transters Accumulator to register Y	impl	0xF0
Flags: - - - - - - T -		
notes		

MYA	Addressing	Opcode
Transters register Y to Accumulator	impl	0xF1
Flags: - - - - - - - -		
notes		

DEY	Addressing	Opcode
Decrements register Y	impl	0x03
Flags: - - - - - T - -		
notes		

INCY	Addressing	Opcode
Increments register Y	impl	0x04
Flags: - - - - - T - -		
notes		

LODS	Addressing	Opcode
Loads Memory into Stackpointer	#	0x20
	abs	0x30
Flags: - - - - - - - -	abs,X	0x40
notes	abs,Y	0x50
	(ind)	0x60
	(ind,X)	0x70

STOS	Addressing	Opcode
Stores Stackpointer into Memory	abs	0x6A
	abs,X	0x7A
Flags: - - - - - - - -	abs,Y	0x8A
notes	(ind)	0x9A
	(ind,X)	0xAA

CSA	Addressing	Opcode
Transters Status register to Accumulator	impl	0xF2
Flags: - - - - - - - -		
notes		

PUSH	Addressing	Opcode
Pushes Register onto the Stack	A	0xBE
	B	0xCE
Flags: - - - - - - - -	FL	0xDE
notes	L	0xEE
	H	0xFE

POP	Addressing	Opcode
Pop the top of the Stack into the Register	A	0xBF
	B	0xCF
Flags: - - - - - - - -	FL	0xDF
notes	L	0xEF
	H	0xFF

LX	Addressing	Opcode
Loads Memory into register pair	LH,#	0x0C
	LH,#	0x0D
Flags: - - - - - - - -		
notes		

JMP	Addressing	Opcode
Loads Memory into ProgramCounter	abs	0x10
Flags: - - - - - - - -		
notes		

ABA	Addressing	Opcode
Adds Accumulator B into Accumulator A	impl	0xF3
Flags: - - - - - T T T		
notes		

SBA	Addressing	Opcode
Subtracts Accumulator B from Accumulator A	impl	0xF4
Flags: - - - - - T T T		
notes		

AAB	Addressing	Opcode
Adds Accumulator A into Accumulator B	impl	0xF5
Flags: - - - - - T T T		
notes		

SAB	Addressing	Opcode
Subtracts Accumulator A from Accumulator B	impl	0xF6
Flags: - - - - - T T T		
notes		

MVI	Addressing	Opcode
Loads Memory into register	L,#	0x1C
	H,#	0x1D
Flags: - - - - - - - -		
notes		

ADCP	Addressing	Opcode
Adds register pair into Accumulator pair	A-L	0xF7
Flags: - - - - - T T T		
notes		

SBCP	Addressing	Opcode
Subtracts register pair into Accumulator pair	A-L	0xF8
Flags: - - - - - T T T		
notes		

XCHG	Addressing	Opcode
Swaps the registers contents	A-L	0xF9
Flags: - - - - - - - -		
notes		

JSR	Addressing	Opcode
Jump to subroutine	abs	0x21
Flags: - - - - - - - -		
notes		

RET	Addressing	Opcode
Return from subroutine	impl	0x4C
Flags: - - - - - - - -		
notes		

JCC	Addressing	Opcode
Jump on Carry clear	abs	0x11
Flags: - - - - - - - -		
notes		

JCS	Addressing	Opcode
Jump on Carry set	abs	0x12
Flags: - - - - - - - -		
notes		

JNE	Addressing	Opcode
Jump on result not Zero	abs	0x13
Flags: - - - - - - - -		
notes		

JEQ	Addressing	Opcode
Jump on result equal to Zero	abs	0x14
Flags: - - - - - - - -		
notes		

JMI	Addressing	Opcode
Jump on negative result	abs	0x15
Flags: - - - - - - - -		
notes		

JPL	Addressing	Opcode
Jump on positive result	abs	0x16
Flags: - - - - - - - -		
notes		

JHI	Addressing	Opcode
Jump on result same or lower	abs	0x17
Flags: - - - - - - - -		
notes		

JLE	Addressing	Opcode
Jump on result higher	abs	0x18
Flags: - - - - - - - -		
notes		

CCC	Addressing	Opcode
Call on Carry clear	abs	0x22
Flags: - - - - - - - -		
notes		

CCS	Addressing	Opcode
Call on Carry set	abs	0x23
Flags: - - - - - - - -		
notes		

CNE	Addressing	Opcode
Call on result not Zero	abs	0x24
Flags: - - - - - - - -		
notes		

CEQ	Addressing	Opcode
Call on result equal to Zero	abs	0x25
Flags: - - - - - - - -		
notes		

CMI	Addressing	Opcode
Call on negative result	abs	0x26
Flags: - - - - - - - -		
notes		

CPL	Addressing	Opcode
Call on positive result	abs	0x27
Flags: - - - - - - - -		
notes		

CHI	Addressing	Opcode
Call on result same or lower	abs	0x28
Flags: - - - - - - - -		
notes		

CLE	Addressing	Opcode
Call on result higher	abs	0x29
Flags: - - - - - - - -		
notes		

CLC	Addressing	Opcode
Clear Carry flag	impl	0x05
Flags: - - - - - - - 0		
notes		

STC	Addressing	Opcode
Set Carry flag	impl	0x06
Flags: - - - - - - - 1		
notes		

CLI	Addressing	Opcode
Clear Interupt flag	impl	0x07
Flags: - - - 0 - - - -		
notes		

STI	Addressing	Opcode
Set Interupt flag	impl	0x08
Flags: - - - 1 - - - -		
notes		

NOP	Addressing	Opcode
No operation	impl	0x2C
Flags: - - - - - - - -		
notes		

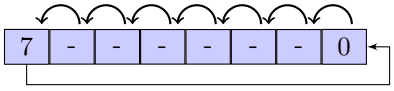
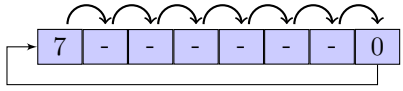
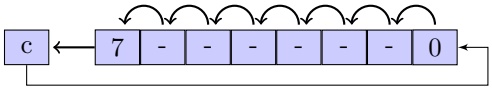
SWI	Addressing	Opcode
Software interupt	impl	0x5C
Flags: - - - 1 - - - -		
Pushes: Accumulators (A then B) Staus register General purpose registers (in order)		

HLT	Addressing	Opcode
Wait for interupt	impl	0x2D
Flags: - - - - - - - -		
notes		

RTI	Addressing	Opcode
Return from software interupt	impl	0x5D
Flags: - - - - - - - -		
Pops: General purpose registers (in order) Staus register Accumulators (B then A)		

			Registers Source							
op	details	Dest	A	B	FL	L	H	M	Stack	FLags
ADC	A + CF + R	A	-	-	-	0x31	0x41	0x51	-	
		B	-	-	-	0x61	0x71	0x81	-	- - - - - T T T
SBC	A - CF - R	A	-	-	-	0x32	0x42	0x52	-	
		B	-	-	-	0x62	0x72	0x82	-	- - - - - T T T
ADD	A + R	A	-	-	-	0x33	0x43	0x53	-	
		B	-	-	-	0x63	0x73	0x83	-	- - - - - T T T
SUB	A - R	A	-	-	-	0x34	0x44	0x54	-	
		B	-	-	-	0x64	0x74	0x84	-	- - - - - T T T
CMP	A - R		-	-	-	0x35	0x45	0x55	-	
			-	-	-	0x65	0x75	0x85	-	- - - - - T T T
OR	A |R	A	-	-	-	0x36	0x46	0x56	-	
		B	-	-	-	0x66	0x76	0x86	-	- - - - - T T -
AND	A & R	A	-	-	-	0x37	0x47	0x57	-	
		B	-	-	-	0x67	0x77	0x87	-	- - - - - T T -
XOR	A (+) R	A	-	-	-	0x38	0x48	0x58	-	
		B	-	-	-	0x68	0x78	0x88	-	- - - - - T T -
BIT	A & R		-	-	-	0x39	0x49	0x59	-	
			-	-	-	0x69	0x79	0x89	-	- - - - - T T -
INCA	A + 1	A	0xD0	-	-	-	-	-	-	- - - - - T T -
INCB	B + 1	B	-	0xE0	-	-	-	-	-	- - - - - T T -
DECA	A - 1	A	0xD1	-	-	-	-	-	-	- - - - - T T -
DECB	B - 1	B	-	0xE1	-	-	-	-	-	- - - - - T T -
RRCA	fig 7	A	0xD2	-	-	-	-	-	-	- - - - - T T T
RRCB	fig 7	B	-	0xE2	-	-	-	-	-	- - - - - T T T
RLCA	fig 6	A	0xD3	-	-	-	-	-	-	- - - - - T T T
RLCB	fig 6	B	-	0xE3	-	-	-	-	-	- - - - - T T T
SALA	fig 1	A	0xD4	-	-	-	-	-	-	- - - - - T T T
SALB	fig 1	B	-	0xE4	-	-	-	-	-	- - - - - T T T
SARA	fijg 2	A	0xD5	-	-	-	-	-	-	- - - - - T T T
SARB	fijg 2	B	-	0xE5	-	-	-	-	-	- - - - - T T T
LSRA	fig 3	A	0xD6	-	-	-	-	-	-	- - - - - T T T
LSRB	fig 3	B	-	0xE6	-	-	-	-	-	- - - - - T T T
COMA	A ∼	A	0xD7	-	-	-	-	-	-	- - - - - T T T
COMB	B ∼	B	-	0xE7	-	-	-	-	-	- - - - - T T T
ROLA	fig 5	A	0xD8	-	-	-	-	-	-	- - - - - T T -
ROLB	fig 5	B	-	0xE8	-	-	-	-	-	- - - - - T T -
RRA	fig 4	A	0xD9	-	-	-	-	-	-	- - - - - T T -
RRB	fig 4	B	-	0xE9	-	-	-	-	-	- - - - - T T -
MOVE	A	A	0x6B	0x7B	-	0x8B	0x9B	0xAB	-	
		B	0x6C	0x7C	-	0x8C	0x9C	0xAC	-	
		L	0x6D	0x7D	-	0x8D	0x9D	0xAD	-	
		H	0x6E	0x7E	-	0x8E	0x9E	0xAE	-	
		M	0x6F	0x7F	-	0x8F	0x9F	-	-	
			-	-	-	-	-	-	-	- - - - - - - -
PUSH	A -*		0xBE	0xCE	0xDE	0xEE	0xFE	-	-	- - - - - - - -
POP	+*	A	-	-	-	-	-	-	0xBF	
		B	-	-	-	-	-	-	0xCF	
		FL	-	-	-	-	-	-	0xDF	
		L	-	-	-	-	-	-	0xEF	
		H	-	-	-	-	-	-	0xFF	- - - - - - - -
ADCP	R + A	A	-	-	-	0xF7	-	-	-	- - - - - T T T
SBCP	R - A	A	-	-	-	0xF8	-	-	-	- - - - - T T T
XCHG	R	A	-	-	-	0xF9	-	-	-	- - - - - - - -

op	details	Dest	#	impl	abs	abs,X	abs,Y	(ind)	(ind,X)	rel	FLags
LDAA	M	A	0x0A	-	0x1A	0x2A	0x3A	0x4A	0x5A	-	- - - - - - - -
LDAB	M	B	0x0B	-	0x1B	0x2B	0x3B	0x4B	0x5B	-	- - - - - - - -
STORA	A	M	-	-	0xBA	0xCA	0xDA	0xEA	0xFA	-	- - - - - - - -
STORB	B	M	-	-	0xBB	0xCB	0xDB	0xEB	0xFB	-	- - - - - - - -
SBIA	A - CF - M	A	0x93	-	-	-	-	-	-	-	- - - - - T T T
SBIB	B - CF - B	B	0x94	-	-	-	-	-	-	-	- - - - - T T T
CPIA	A - M		0x95	-	-	-	-	-	-	-	- - - - - T T T
CPIB	B - B		0x96	-	-	-	-	-	-	-	- - - - - T T T
ORIA	A |M	A	0x97	-	-	-	-	-	-	-	- - - - - T T -
ORIB	B |B	B	0x98	-	-	-	-	-	-	-	- - - - - T T -
INC	M + 1	M	-	-	0xA0	0xB0	0xC0	-	-	-	- - - - - T T -
DEC	M - 1	M	-	-	0xA1	0xB1	0xC1	-	-	-	- - - - - T T -
RRC	fig 7	M	-	-	0xA2	0xB2	0xC2	-	-	-	- - - - - T T T
RLC	fig 6	M	-	-	0xA3	0xB3	0xC3	-	-	-	- - - - - T T T
SAL	fig 1	M	-	-	0xA4	0xB4	0xC4	-	-	-	- - - - - T T T
SAR	fijg 2	M	-	-	0xA5	0xB5	0xC5	-	-	-	- - - - - T T T
LSR	fig 3	M	-	-	0xA6	0xB6	0xC6	-	-	-	- - - - - T T T
COM	M ∼	M	-	-	0xA7	0xB7	0xC7	-	-	-	- - - - - T T T
ROL	fig 5	M	-	-	0xA8	0xB8	0xC8	-	-	-	- - - - - T T -
RR	fig 4	M	-	-	0xA9	0xB9	0xC9	-	-	-	- - - - - T T -
LDX	M	X	0x0E	-	0x1E	0x2E	0x3E	0x4E	0x5E	-	- - - - - - - -
STOX	X	M	-	-	0xBC	0xCC	0xDC	0xEC	0xFC	-	- - - - - - - -
DECX	X - 1	X	-	0x01	-	-	-	-	-	-	- - - - - T - -
INCX	X + 1	X	-	0x02	-	-	-	-	-	-	- - - - - T - -
LDY	M	Y	0x0F	-	0x1F	0x2F	0x3F	0x4F	0x5F	-	- - - - - - - -
STOY	Y	M	-	-	0xBD	0xCD	0xDD	0xED	0xFD	-	- - - - - - - -
CAY	A	Y	-	0xF0	-	-	-	-	-	-	- - - - - - T -
MYA	Y	A	-	0xF1	-	-	-	-	-	-	- - - - - - - -
DEY	Y - 1	Y	-	0x03	-	-	-	-	-	-	- - - - - T - -
INCY	Y + 1	Y	-	0x04	-	-	-	-	-	-	- - - - - T - -
LODS	M	SP	0x20	-	0x30	0x40	0x50	0x60	0x70	-	- - - - - - - -
STOS	SP	M	-	-	0x6A	0x7A	0x8A	0x9A	0xAA	-	- - - - - - - -
CSA	FL	A	-	0xF2	-	-	-	-	-	-	- - - - - - - -
LX		L	0x0D	-	-	-	-	-	-	-	- - - - - - - -
JMP			-	-	0x10	-	-	-	-	-	- - - - - - - -
ABA	A + B	A	-	0xF3	-	-	-	-	-	-	- - - - - T T T
SBA	A - B	A	-	0xF4	-	-	-	-	-	-	- - - - - T T T
AAB	B + A	B	-	0xF5	-	-	-	-	-	-	- - - - - T T T
SAB	B - A	B	-	0xF6	-	-	-	-	-	-	- - - - - T T T
MVI	M	L	0x1C	-	-	-	-	-	-	-	
		H	0x1D	-	-	-	-	-	-	-	- - - - - - - -
JSR			-	-	0x21	-	-	-	-	-	- - - - - - - -
RET			-	0x4C	-	-	-	-	-	-	- - - - - - - -
JCC	CF = 0		-	-	0x11	-	-	-	-	-	- - - - - - - -
JCS	CF = 1		-	-	0x12	-	-	-	-	-	- - - - - - - -
JNE	ZF = 0		-	-	0x13	-	-	-	-	-	- - - - - - - -
JEQ	ZF = 1		-	-	0x14	-	-	-	-	-	- - - - - - - -
JMI	NF = 1		-	-	0x15	-	-	-	-	-	- - - - - - - -
JPL	NF = 0		-	-	0x16	-	-	-	-	-	- - - - - - - -
JHI	CF |ZF = 1		-	-	0x17	-	-	-	-	-	- - - - - - - -
JLE	CF |ZF = 0		-	-	0x18	-	-	-	-	-	- - - - - - - -
CCC	CF = 0		-	-	0x22	-	-	-	-	-	- - - - - - - -
CCS	CF = 1		-	-	0x23	-	-	-	-	-	- - - - - - - -
CNE	ZF = 0		-	-	0x24	-	-	-	-	-	- - - - - - - -
CEQ	ZF = 1		-	-	0x25	-	-	-	-	-	- - - - - - - -
CMI	NF = 1		-	-	0x26	-	-	-	-	-	- - - - - - - -
CPL	NF = 0		-	-	0x27	-	-	-	-	-	- - - - - - - -
CHI	CF |ZF = 1		-	-	0x28	-	-	-	-	-	- - - - - - - -
CLE	CF |ZF = 0		-	-	0x29	-	-	-	-	-	- - - - - - - -
CLC	CF = 0		-	0x05	-	-	-	-	-	-	- - - - - - - 0
STC	CF = 1		-	0x06	-	-	-	-	-	-	- - - - - - - 1
CLI	IF = 0		-	0x07	-	-	-	-	-	-	- - - 0 - - - -
STI	IF = 1		-	0x08	-	-	-	-	-	-	- - - 1 - - - -
NOP			-	0x2C	-	-	-	-	-	-	- - - - - - - -
HLT			-	0x2D	-	-	-	-	-	-	- - - - - - - -
SWI	-		-	0x5C	-	-	-	-	-	-	- - - 1 - - - -
RTI	-		-	0x5D	-	-	-	-	-	-	- - - - - - - -

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