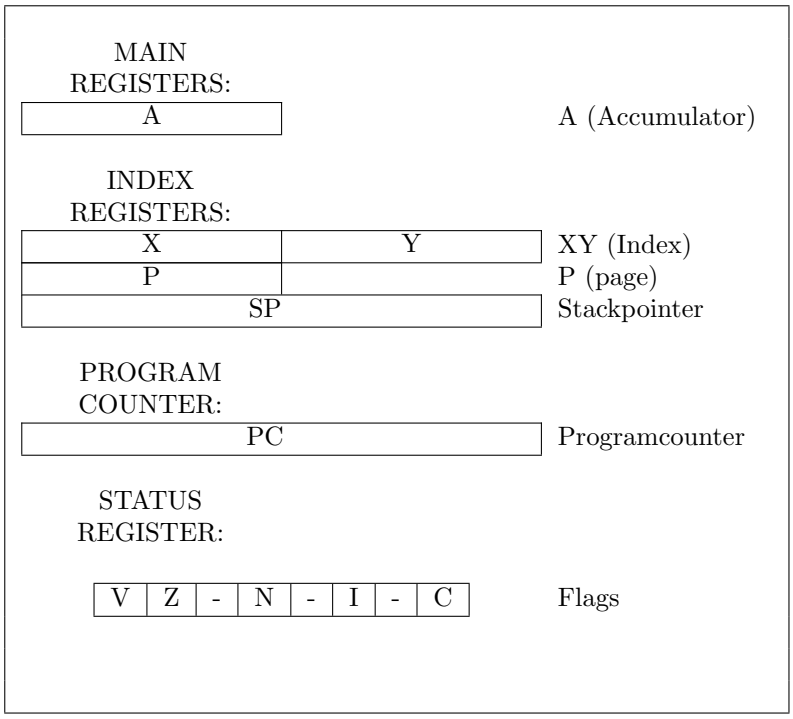
Chimera-2017V

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 or Register Y or Register XY). 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.

PAGED ADDRESSING(pag)

In paged addressing, the second byte of the instruction represents the low order byte of the effective address. The content of the page register represents the high order byte of the effective address. Using both in conjunction give the effective address.

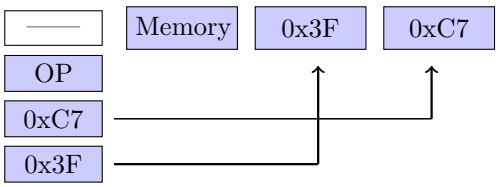
OFFSET ADDRESSING(rel)

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.


Memory
0x3F
0xC7
OP
0xC7
0x3F

Hexadecimal Matrix

-	-	LOW NIBBLE															
-	-	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
HIGH NIBBLE	0x0	-	CLC	-	LDA	LDA	LDA	LDA	LDA	LDA	-	STO	STO	STO	STO	STO	-
	-	-	impl	-	#	abs	abs,X	abs,Y	abs,XY	pag	-	abs	abs,X	abs,Y	abs,XY	pag	-
	0x1	TAY	STC	-	ADC	ADC	ADC	ADC	ADC	ADC	TST	TST	TST	TST	TSTA	PSHA	PSHs
	-	impl	impl	-	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	impl
	0x2	TYA	CLI	-	SBC	SBC	SBC	SBC	SBC	SBC	INC	INC	INC	INC	INCA	POPA	POPs
	-	impl	impl	-	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	impl
	0x3	TSA	STI	-	ADD	ADD	ADD	ADD	ADD	ADD	DEC	DEC	DEC	DEC	DECA	-	JMPR
	-	impl	impl	-	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	-	abs
	0x4	TAP	SEV	DECX	SUB	SUB	SUB	SUB	SUB	SUB	RCR	RCR	RCR	RCR	RCRA	SRA	CCC
	-	impl	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0x5	TPA	CLV	INCX	CMP	CMP	CMP	CMP	CMP	CMP	RCL	RCL	RCL	RCL	RCLA	SCC	CCS
	-	impl	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0x6	-	CMC	DEY	OR	OR	OR	OR	OR	OR	SHL	SHL	SHL	SHL	SHLA	SCS	CNE
	-	-	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0x7	JP	CMV	INY	AND	AND	AND	AND	AND	AND	SAR	SAR	SAR	SAR	SARA	SNE	CEQ
	-	abs	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0x8	JCC	SWI	DEP	EOR	EOR	EOR	EOR	EOR	EOR	LSR	LSR	LSR	LSR	LSRA	SEQ	CVC
	-	abs	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0x9	JCS	RTI	INP	BT	BT	BT	BT	BT	BT	NOT	NOT	NOT	NOT	NOTA	SVC	CVS
	-	abs	impl	impl	#	abs	abs,X	abs,Y	abs,XY	pag	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0xa	JNE	LDX	LDX	LDX	LDX	LDX	LDX	RTS	-	NEG	NEG	NEG	NEG	NEGA	SVS	CMI
	-	abs	#	abs	abs,X	abs,Y	abs,XY	pag	impl	-	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0xb	JEQ	LDY	LDY	LDY	LDY	LDY	LDY	-	-	RAL	RAL	RAL	RAL	RALA	SMI	CPL
	-	abs	#	abs	abs,X	abs,Y	abs,XY	pag	-	-	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0xc	JVC	LDP	LDP	LDP	LDP	LDP	LDP	-	-	ROR	ROR	ROR	ROR	RORA	SPL	CHI
	-	abs	#	abs	abs,X	abs,Y	abs,XY	pag	-	-	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0xd	JVS	-	-	STX	STX	STX	STX	STX	-	CLR	CLR	CLR	CLR	CLRA	SGE	CLE
	-	abs	-	-	abs	abs,X	abs,Y	abs,XY	pag	-	abs	abs,X	abs,Y	abs,XY	A	impl	abs
	0xe	JMI	NOP	-	STY	STY	STY	STY	STY	LODS	LODS	LODS	LODS	LODS	LODS	SGT	-
	-	abs	impl	-	abs	abs,X	abs,Y	abs,XY	pag	#	abs	abs,X	abs,Y	abs,XY	pag	impl	-
	0xf	JPL	HALT	-	STP	STP	STP	STP	STP	-	-	-	STOS	STOS	STOS	STOS	STOS
	-	abs	impl	-	abs	abs,X	abs,Y	abs,XY	pag	-	-	-	abs	abs,X	abs,Y	abs,XY	pag

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

LDA		Addressing	Opcode
Loads Memory into Accumulator		#	0x03
Flags:	- T - T - - - 0	abs	0x04
notes		abs,X	0x05
		abs,Y	0x06
		abs,XY	0x07
		pag	0x08

STO		Addressing	Opcode
Stores Accumulator into Memory		abs	0x0A
Flags:	- T - T - - - 0	abs,X	0x0B
notes		abs,Y	0x0C
		abs,XY	0x0D
		pag	0x0E

ADC		Addressing	Opcode
Memory added to Accumulator with Carry		#	0x13
Flags:	T T - T - - - T	abs	0x14
notes		abs,X	0x15
		abs,Y	0x16
		abs,XY	0x17
		pag	0x18

SBC		Addressing	Opcode
Memory subtracted to Accumulator with Carry		#	0x23
Flags:	T T - T - - - T	abs	0x24
notes		abs,X	0x25
		abs,Y	0x26
		abs,XY	0x27
		pag	0x28

ADD		Addressing	Opcode
Memory added to Accumulator		#	0x33
Flags:	T T - T - - - T	abs	0x34
notes		abs,X	0x35
		abs,Y	0x36
		abs,XY	0x37
		pag	0x38

SUB		Addressing	Opcode
Memory subtracted to Accumulator		#	0x43
Flags:	T T - T - - - T	abs	0x44
notes		abs,X	0x45
		abs,Y	0x46
		abs,XY	0x47
		pag	0x48

CMP		Addressing	Opcode
Memory compared to Accumulator		#	0x53
Flags:	T T - T - - - T	abs	0x54
notes		abs,X	0x55
		abs,Y	0x56
		abs,XY	0x57
		pag	0x58

OR		Addressing	Opcode
Memory bitwise inclusive or with Accumulator		#	0x63
Flags:	- T - T - - - -	abs	0x64
notes		abs,X	0x65
		abs,Y	0x66
		abs,XY	0x67
		pag	0x68

AND		Addressing	Opcode
Memory bitwise and with Accumulator		#	0x73
Flags:	- T - T - - - -	abs	0x74
notes		abs,X	0x75
		abs,Y	0x76
		abs,XY	0x77
		pag	0x78

EOR		Addressing	Opcode
Memory bitwise exclusive or with Accumulator		#	0x83
Flags:	- T - T - - - -	abs	0x84
notes		abs,X	0x85
		abs,Y	0x86
		abs,XY	0x87
		pag	0x88

BT		Addressing	Opcode
Memory Bit tested with Accumulator		#	0x93
Flags:	- T - T - - - -	abs	0x94
notes		abs,X	0x95
		abs,Y	0x96
		abs,XY	0x97
		pag	0x98

TST		Addressing	Opcode
Bit test Memory or Accumulator		abs	0x19
Flags:	- T - T - - - -	abs,X	0x1A
notes		abs,Y	0x1B
		abs,XY	0x1C

TSTA	Addressing	Opcode
Bit test Memory or Accumulator	A	0x1D
Flags: - T - T - - - -		
notes		

INC	Addressing	Opcode
Increment Memory or Accumulator	abs	0x29
	abs,X	0x2A
Flags: - T - T - - - -	abs,Y	0x2B
notes	abs,XY	0x2C

INCA	Addressing	Opcode
Increment Memory or Accumulator	A	0x2D
Flags: - T - T - - - -		
notes		

DEC	Addressing	Opcode
Decrement Memory or Accumulator	abs	0x39
	abs,X	0x3A
Flags: - T - T - - - -	abs,Y	0x3B
notes	abs,XY	0x3C

DECA	Addressing	Opcode
Decrement Memory or Accumulator	A	0x3D
Flags: - T - T - - - -		
notes		

RCR	Addressing	Opcode
Rotate right through carry Memory or Accumulator	abs	0x49
	abs,X	0x4A
	abs,Y	0x4B
Flags: - T - T - - - T	abs,XY	0x4C
notes		

RCRA	Addressing	Opcode
Rotate right through carry Memory or Accumulator	A	0x4D
Flags: - T - T - - - T		
notes		

RCL	Addressing	Opcode
Rotate left through carry Memory or Accumulator	abs	0x59
	abs,X	0x5A
	abs,Y	0x5B
Flags: - T - T - - - T	abs,XY	0x5C
notes		

RCLA	Addressing	Opcode
Rotate left through carry Memory or Accumulator	A	0x5D
Flags: - T - T - - - T		
notes		

SHL	Addressing	Opcode
Arithmetic shift left Memory or Accumulator	abs	0x69
	abs,X	0x6A
	abs,Y	0x6B
Flags: - T - T - - - T	abs,XY	0x6C
notes		

SHLA	Addressing	Opcode
Arithmetic shift left Memory or Accumulator	A	0x6D
Flags: - T - T - - - T		
notes		

SAR	Addressing	Opcode
Arithmetic shift right Memory or Accumulator	abs	0x79
	abs,X	0x7A
	abs,Y	0x7B
Flags: - T - T - - - T	abs,XY	0x7C
notes		

SARA	Addressing	Opcode
Arithmetic shift right Memory or Accumulator	A	0x7D
Flags: - T - T - - - T		
notes		

LSR	Addressing	Opcode
Shift right Memory or Accumulator	abs	0x89
	abs,X	0x8A
Flags: - T - T - - - T	abs,Y	0x8B
notes	abs,XY	0x8C

LSRA	Addressing	Opcode
Shift right Memory or Accumulator	A	0x8D
Flags: - T - T - - - T		
notes		

NOT	Addressing	Opcode
Negate Memory or Accumulator	abs	0x99
	abs,X	0x9A
Flags: - T - T - - - T	abs,Y	0x9B
notes	abs,XY	0x9C

NOTA	Addressing	Opcode
Negate Memory or Accumulator	A	0x9D
Flags: - T - T - - - T		
notes		

NEG	Addressing	Opcode
2's complement Memory or Accumulator	abs	0xA9
	abs,X	0xAA
	abs,Y	0xAB
Flags: - T - T - - - -	abs,XY	0xAC
notes		

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

RAL	Addressing	Opcode
Rotate left without carry Memory or Accumulator	abs	0xB9
	abs,X	0xBA
	abs,Y	0xBB
Flags: - T - T - - - -	abs,XY	0xBC
notes		

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

ROR	Addressing	Opcode
Rotate right without carry Memory or Accumulator	abs	0xC9
	abs,X	0xCA
	abs,Y	0xCB
Flags: - T - T - - - -	abs,XY	0xCC
notes		

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

CLR	Addressing	Opcode
Clear Memory or Accumulator	abs	0xD9
	abs,X	0xDA
Flags: - 1 - 0 - - - -	abs,Y	0xDB
notes	abs,XY	0xDC

CLRA	Addressing	Opcode
Clear Memory or Accumulator	A	0xDD
Flags: - 1 - 0 - - - -		
notes		

LDX	Addressing	Opcode
Loads Memory into register X	#	0xA1
	abs	0xA2
Flags: - T - T - - - 0	abs,X	0xA3
notes	abs,Y	0xA4
	abs,XY	0xA5
	pag	0xA6

STX	Addressing	Opcode
Stores register X into Memory	abs	0xD3
	abs,X	0xD4
Flags: - T - T - - - 0	abs,Y	0xD5
notes	abs,XY	0xD6
	pag	0xD7

DECX	Addressing	Opcode
Decrements register X	impl	0x42
Flags: - T - - - - - -		
notes		

INCX	Addressing	Opcode
Increments register X	impl	0x52
Flags: - T - - - - - -		
notes		

LDY	Addressing	Opcode
Loads Memory into register Y	#	0xB1
	abs	0xB2
Flags: - T - T - - - 0	abs,X	0xB3
notes	abs,Y	0xB4
	abs,XY	0xB5
	pag	0xB6

STY	Addressing	Opcode
Stores register Y into Memory	abs	0xE3
Flags: - T - T - - - 0	abs,X	0xE4
notes	abs,Y	0xE5
	abs,XY	0xE6
	pag	0xE7

TAY	Addressing	Opcode
Transters Accumulator to register Y	impl	0x10
Flags: - - - T - - - -		
notes		

TYA	Addressing	Opcode
Transters register Y to Accumulator	impl	0x20
Flags: - T - T - - - -		
notes		

DEY	Addressing	Opcode
Decrements register Y	impl	0x62
Flags: - T - - - - - -		
notes		

INY	Addressing	Opcode
Increments register Y	impl	0x72
Flags: - T - - - - - -		
notes		

LODS	Addressing	Opcode
Loads Memory into Stackpointer	#	0xE8
Flags: - T - T - - - 0	abs	0xE9
notes	abs,X	0xEA
	abs,Y	0xEB
	abs,XY	0xEC
	pag	0xED

STOS	Addressing	Opcode
Stores Stackpointer into Memory	abs	0xFB
Flags: - T - T - - - 0	abs,X	0xFC
notes	abs,Y	0xFD
	abs,XY	0xFE
	pag	0xFF

TSA	Addressing	Opcode
Transters Status register to Accumulator	impl	0x30
Flags: - - - - - - - -		
notes		

PSHA	Addressing	Opcode
Pushes Register onto the Stack	impl	0x1E
Flags: - - - - - - - -		
notes		

PSHs	Addressing	Opcode
Pushes Register onto the Stack	impl	0x1F
Flags: - - - - - - - -		
notes		

POPA	Addressing	Opcode
Pop the top of the Stack into the Register	impl	0x2E
Flags: - - - - - - - -		
notes		

POPs	Addressing	Opcode
Pop the top of the Stack into the Register	impl	0x2F
Flags: - - - - - - - -		
notes		

JP	Addressing	Opcode
Loads Memory into ProgramCounter	abs	0x70
Flags: - - - - - - - -		
notes		

JMPR	Addressing	Opcode
Jump to subroutine	abs	0x3F
Flags: - - - - - - - -		
notes		

RTS	Addressing	Opcode
Return from subroutine	impl	0xA7
Flags: - - - - - - - -		
notes		

SRA	Addressing	Opcode
Skip always	impl	0x4E
Flags: - - - - - - - -		
notes		

SCC	Addressing	Opcode
Skip on Carry clear	impl	0x5E
Flags: - - - - - - - -		
notes		

SCS	Addressing	Opcode
Skip on Carry set	impl	0x6E
Flags: - - - - - - - -		
notes		

SNE	Addressing	Opcode
Skip on result not Zero	impl	0x7E
Flags: - - - - - - - -		
notes		

SEQ	Addressing	Opcode
Skip on result equal to Zero	impl	0x8E
Flags: - - - - - - - -		
notes		

SVC	Addressing	Opcode
Skip on overflow clear	impl	0x9E
Flags: - - - - - - - -		
notes		

SVS	Addressing	Opcode
Skip on overflow set	impl	0xAE
Flags: - - - - - - - -		
notes		

SMI	Addressing	Opcode
Skip on negative result	impl	0xBE
Flags: - - - - - - - -		
notes		

SPL	Addressing	Opcode
Skip on positive result	impl	0xCE
Flags: - - - - - - - -		
notes		

SGE	Addressing	Opcode
Skip on result less then or equal to zero	impl	0xDE
Flags: - - - - - - - -		
notes		

SGT	Addressing	Opcode
Skip on result greater then or equal to zero	impl	0xEE
Flags: - - - - - - - -		
notes		

JCC	Addressing	Opcode
Jump on Carry clear	abs	0x80
Flags: - - - - - - - -		
notes		

JCS	Addressing	Opcode
Jump on Carry set	abs	0x90
Flags: - - - - - - - -		
notes		

JNE	Addressing	Opcode
Jump on result not Zero	abs	0xA0
Flags: - - - - - - - -		
notes		

JEQ	Addressing	Opcode
Jump on result equal to Zero	abs	0xB0
Flags: - - - - - - - -		
notes		

JVC	Addressing	Opcode
Jump on overflow clear	abs	0xC0
Flags: - - - - - - - -		
notes		

JVS	Addressing	Opcode
Jump on overflow set	abs	0xD0
Flags: - - - - - - - -		
notes		

JMI	Addressing	Opcode
Jump on negative result	abs	0xE0
Flags: - - - - - - - -		
notes		

JPL	Addressing	Opcode
Jump on positive result	abs	0xF0
Flags: - - - - - - - -		
notes		

CCC	Addressing	Opcode
Call on Carry clear	abs	0x4F
Flags: - - - - - - - -		
notes		

CCS	Addressing	Opcode
Call on Carry set	abs	0x5F
Flags: - - - - - - - -		
notes		

CNE	Addressing	Opcode
Call on result not Zero	abs	0x6F
Flags: - - - - - - - -		
notes		

CEQ	Addressing	Opcode
Call on result equal to Zero	abs	0x7F
Flags: - - - - - - - -		
notes		

CVC	Addressing	Opcode
Call on overflow clear	abs	0x8F
Flags: - - - - - - - -		
notes		

CVS	Addressing	Opcode
Call on overflow set	abs	0x9F
Flags: - - - - - - - -		
notes		

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

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

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

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

CLC	Addressing	Opcode
Clear Carry flag	impl	0x01
Flags: - - - - - - - 0		
notes		

STC	Addressing	Opcode
Set Carry flag	impl	0x11
Flags: - - - - - - - 1		
notes		

CLI	Addressing	Opcode
Clear Interupt flag	impl	0x21
Flags: - - - - - 0 - -		
notes		

STI	Addressing	Opcode
Set Interupt flag	impl	0x31
Flags: - - - - - 1 - -		
notes		

SEV	Addressing	Opcode
Set Overflow flag	impl	0x41
Flags: 1 - - - - - - -		
notes		

CLV	Addressing	Opcode
Clear Overflow flag	impl	0x51
Flags: 0 - - - - - - -		
notes		

CMC	Addressing	Opcode
Compliment carry flag	impl	0x61
Flags: - - - - - - - -		
notes		

CMV	Addressing	Opcode
Compliment Overflow flag	impl	0x71
Flags: - - - - - - - -		
notes		

NOP	Addressing	Opcode
No operation	impl	0xE1
Flags: - - - - - - - -		
notes		

HALT	Addressing	Opcode
Wait for interupt	impl	0xF1
Flags: - - - - - - - -		
notes		

SWI	Addressing	Opcode
Software interupt	impl	0x81
Flags: - - - - - - - -		
Pushes: Accumulator Staus register		

RTI	Addressing	Opcode
Return from software interupt	impl	0x91
Flags: - - - - - - - -		
Pops: Staus register Accumulator		

LDP	Addressing	Opcode
Loads Memory into register P	#	0xC1
	abs	0xC2
Flags: - - - - - - - -	abs,X	0xC3
notes	abs,Y	0xC4
	abs,XY	0xC5
	pag	0xC6

STP	Addressing	Opcode
Stores register P into Memory	abs	0xF3
	abs,X	0xF4
Flags: - - - - - - - -	abs,Y	0xF5
notes	abs,XY	0xF6
	pag	0xF7

TAP	Addressing	Opcode
Transters Accumulator to register P	impl	0x40
Flags: - - - - - - - -		
notes		

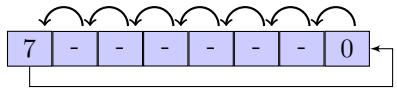
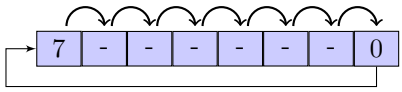
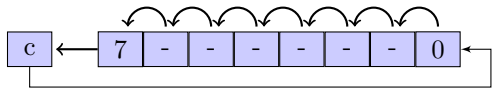
TPA	Addressing	Opcode
Transters register P to Accumulator	impl	0x50
Flags: - - - - - - - -		
notes		

DEP		Addressing	Opcode
Decrements Page register		impl	0x82
Flags:	- T - - - - - -		
notes			

INP		Addressing	Opcode
Increments Page register		impl	0x92
Flags:	- T - - - - - -		
notes			

			Registers Source			
op	details	Dest	A	FL	Stack	FLags
TSTA	A - 0	A	0x1D	-	-	- T - T - - - -
INCA	A + 1	A	0x2D	-	-	- T - T - - - -
DECA	A - 1	A	0x3D	-	-	- T - T - - - -
RCRA	fig 7	A	0x4D	-	-	- T - T - - - T
RCLA	fig 6	A	0x5D	-	-	- T - T - - - T
SHLA	fig 1	A	0x6D	-	-	- T - T - - - T
SARA	fijg 2	A	0x7D	-	-	- T - T - - - T
LSRA	fig 3	A	0x8D	-	-	- T - T - - - T
NOTA	A $\sim$	A	0x9D	-	-	- T - T - - - T
NEGA	0 - A	A	0xAD	-	-	- T - T - - - -
RALA	fig 5	A	0xBD	-	-	- T - T - - - -
RORA	fig 4	A	0xCD	-	-	- T - T - - - -
CLRA	0	A	-	-	-	- 1 - 0 - - - -

op	details	Dest	#	impl	abs	abs,X	abs,Y	abs,XY	pag	rel	FLags
LDA	M	A	0x03	-	0x04	0x05	0x06	0x07	0x08	-	- T - T - - - 0
STO	A	M	-	-	0x0A	0x0B	0x0C	0x0D	0x0E	-	- T - T - - - 0
ADC	M + A + CF	A	0x13	-	0x14	0x15	0x16	0x17	0x18	-	T T - T - - - T
SBC	A - CF - M	A	0x23	-	0x24	0x25	0x26	0x27	0x28	-	T T - T - - - T
ADD	M + A	A	0x33	-	0x34	0x35	0x36	0x37	0x38	-	T T - T - - - T
SUB	A - M	A	0x43	-	0x44	0x45	0x46	0x47	0x48	-	T T - T - - - T
CMP	A - M		0x53	-	0x54	0x55	0x56	0x57	0x58	-	T T - T - - - T
OR	M |A	A	0x63	-	0x64	0x65	0x66	0x67	0x68	-	- T - T - - - -
AND	M & A	A	0x73	-	0x74	0x75	0x76	0x77	0x78	-	- T - T - - - -
EOR	M (+) A	A	0x83	-	0x84	0x85	0x86	0x87	0x88	-	- T - T - - - -
BT	M & A		0x93	-	0x94	0x95	0x96	0x97	0x98	-	- T - T - - - -
TST	M - 0		-	-	0x19	0x1A	0x1B	0x1C	-	-	- T - T - - - -
INC	M + 1	M	-	-	0x29	0x2A	0x2B	0x2C	-	-	- T - T - - - -
DEC	M - 1	M	-	-	0x39	0x3A	0x3B	0x3C	-	-	- T - T - - - -
RCR	fig 7	M	-	-	0x49	0x4A	0x4B	0x4C	-	-	- T - T - - - T
RCL	fig 6	M	-	-	0x59	0x5A	0x5B	0x5C	-	-	- T - T - - - T
SHL	fig 1	M	-	-	0x69	0x6A	0x6B	0x6C	-	-	- T - T - - - T
SAR	fijg 2	M	-	-	0x79	0x7A	0x7B	0x7C	-	-	- T - T - - - T
LSR	fig 3	M	-	-	0x89	0x8A	0x8B	0x8C	-	-	- T - T - - - T
NOT	M ∼	M	-	-	0x99	0x9A	0x9B	0x9C	-	-	- T - T - - - T
NEG	0 - M	M	-	-	0xA9	0xAA	0xAB	0xAC	-	-	- T - T - - - -
RAL	fig 5	M	-	-	0xB9	0xBA	0xBB	0xBC	-	-	- T - T - - - -
ROR	fig 4	M	-	-	0xC9	0xCA	0xCB	0xCC	-	-	- T - T - - - -
CLR	0	M	-	-	0xD9	0xDA	0xDB	0xDC	-	-	- 1 - 0 - - - -
LDX	M	X	0xA1	-	0xA2	0xA3	0xA4	0xA5	0xA6	-	- T - T - - - 0
STX	X	M	-	-	0xD3	0xD4	0xD5	0xD6	0xD7	-	- T - T - - - 0
DECX	X - 1	X	-	0x42	-	-	-	-	-	-	- T - - - - - -
INCX	X + 1	X	-	0x52	-	-	-	-	-	-	- T - - - - - -
LDY	M	Y	0xB1	-	0xB2	0xB3	0xB4	0xB5	0xB6	-	- T - T - - - 0
STY	Y	M	-	-	0xE3	0xE4	0xE5	0xE6	0xE7	-	- T - T - - - 0
TAY	A	Y	-	0x10	-	-	-	-	-	-	- - - T - - - -
TYA	Y	A	-	0x20	-	-	-	-	-	-	- T - T - - - -
DEY	Y - 1	Y	-	0x62	-	-	-	-	-	-	- T - - - - - -
INY	Y + 1	Y	-	0x72	-	-	-	-	-	-	- T - - - - - -
LODS	M	SP	0xE8	-	0xE9	0xEA	0xEB	0xEC	0xED	-	- T - T - - - 0
STOS	SP	M	-	-	0xFB	0xFC	0xFD	0xFE	0xFF	-	- T - T - - - 0
TSA	FL	A	-	0x30	-	-	-	-	-	-	- - - - - - - -
PSHA	A -*		-	0x1E	-	-	-	-	-	-	- - - - - - - -
PSHs	FL -*		-	0x1F	-	-	-	-	-	-	- - - - - - - -
POPA	+*	A	0x2E	-	-	-	-	-	-	-	- - - - - - - -
POPs	+*	S	0x2F	-	-	-	-	-	-	-	- - - - - - - -
JP			-	-	0x70	-	-	-	-	-	- - - - - - - -
JMPR			-	-	0x3F	-	-	-	-	-	- - - - - - - -
RTS			-	0xA7	-	-	-	-	-	-	- - - - - - - -
SRA			-	0x4E	-	-	-	-	-	-	- - - - - - - -
SCC	CF = 0		-	0x5E	-	-	-	-	-	-	- - - - - - - -
SCS	CF = 1		-	0x6E	-	-	-	-	-	-	- - - - - - - -
SNE	ZF = 0		-	0x7E	-	-	-	-	-	-	- - - - - - - -
SEQ	ZF = 1		-	0x8E	-	-	-	-	-	-	- - - - - - - -
SVC	VF = 0		-	0x9E	-	-	-	-	-	-	- - - - - - - -
SVS	VF = 1		-	0xAE	-	-	-	-	-	-	- - - - - - - -
SMI	NF = 1		-	0xBE	-	-	-	-	-	-	- - - - - - - -
SPL	NF = 0		-	0xCE	-	-	-	-	-	-	- - - - - - - -
SGE	NF ˆVF = 0		-	0xDE	-	-	-	-	-	-	- - - - - - - -
SGT	ZF |NF ˆVF = 1		-	0xEE	-	-	-	-	-	-	- - - - - - - -
JCC	CF = 0		-	-	0x80	-	-	-	-	-	- - - - - - - -
JCS	CF = 1		-	-	0x90	-	-	-	-	-	- - - - - - - -
JNE	ZF = 0		-	-	0xA0	-	-	-	-	-	- - - - - - - -
JEQ	ZF = 1		-	-	0xB0	-	-	-	-	-	- - - - - - - -
JVC	VF = 0		-	-	0xC0	-	-	-	-	-	- - - - - - - -
JVS	VF = 1		-	-	0xD0	-	-	-	-	-	- - - - - - - -
JMI	NF = 1		-	-	0xE0	-	-	-	-	-	- - - - - - - -
JPL	NF = 0		-	-	0xF0	-	-	-	-	-	- - - - - - - -
CCC	CF = 0		-	-	0x4F	-	-	-	-	-	- - - - - - - -
CCS	CF = 1		-	-	0x5F	-	-	-	-	-	- - - - - - - -
CNE	ZF = 0		-	-	0x6F	-	-	-	-	-	- - - - - - - -
CEQ	ZF = 1		-	-	0x7F	-	-	-	-	-	- - - - - - - -
CVC	VF = 0		-	-	0x8F	-	-	-	-	-	- - - - - - - -
CVS	VF = 1		-	-	0x9F	-	-	-	-	-	- - - - - - - -
CMI	NF = 1		-	-	0xAF	-	-	-	-	-	- - - - - - - -
CPL	NF = 0		-	-	0xBF	-	-	-	-	-	- - - - - - - -
CHI	CF |ZF = 1		-	-	0xCF	-	-	-	-	-	- - - - - - - -
CLE	CF |ZF = 0		-	-	0xDF	-	-	-	-	-	- - - - - - - -
CLC	CF = 0		-	0x01	-	-	-	-	-	-	- - - - - - - 0
STC	CF = 1		-	0x11	-	-	-	-	-	-	- - - - - - - 1
CLI	IF = 0		-	0x21	-	-	-	-	-	-	- - - - - 0 - -
STI	IF = 1		-	0x31	-	-	-	-	-	-	- - - - - 1 - -
SEV	VF = 1		-	0x41	-	-	-	-	-	-	1 - - - - - - -
CLV	VF = 0		-	0x51	-	-	-	-	-	-	0 - - - - - - -
CMC	CF ∼		-	0x61	-	-	-	-	-	-	- - - - - - - -
CMV	VF ∼		-	0x71	-	-	-	-	-	-	- - - - - - - -
NOP			-	0xE1	-	-	-	-	-	-	- - - - - - - -
HALT			-	0xF1	-	-	-	-	-	-	- - - - - - - -
SWI	-		-	0x81	-	-	-	-	-	-	- - - - - - - -
RTI	-		-	0x91	-	-	-	-	-	-	- - - - - - - -
LDP	M	P	0xC1	-	0xC2	0xC3	0xC4	0xC5	0xC6	-	- - - - - - - -
STP	P	M	-	-	0xF3	0xF4	0xF5	0xF6	0xF7	-	- - - - - - - -
TAP	A	P	-	0x40	-	-	-	-	-	-	- - - - - - - -
TPA	P	A	-	0x50	-	-	-	-	-	-	- - - - - - - -
DEP	P - 1	P	-	0x82	-	-	-	-	-	-	- T - - - - - -
INP	P + 1	P	-	0x92	-	-	-	-	-	-	- T - - - - - -

Key	
A - Accumulator FL - Status Register |- Inclusive or & - logical and -* - Push to stack and decrement stack pointer X - Index Register CF - Carry FLag NF - Negative FLag VF - OverflowFlag FLag P - Page Register	SP - StackPointer M - Memory (+) - Exclusive or $\sim$ - Negation +* - Increment stack pointer and pop from stack ZF - Zero FLag IF - Zero FLag
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