

ADDRESSING MODES SUMMARY

MODE

DESCRIPTION

EXAMPLES

IMMEDIATE

Uses specific data (a byte) as the argument.
Identified by the symbol # at the beginning of the argument

LDA # $\$A5$ (yields $A = \$D5$)
LDX # $\$A5$ (X points to memory location $\$00A5$)

DIRECT

The argument represents an 8-bit memory location.
Obviously, only the first 2^8 memory spots can be addressed this way

LDA $\$A5$ ($A = (\$A5)$)
(yields $A = \$BF$, what's stored in location $\$00A5$)
LDX $\$A5$ ($X = (\$A5)$) \rightsquigarrow X \rightarrow $\$00BF$
(yields $X = \$BF$, meaning X now points to memory spot $\$00BF$)
LDX $\$183A$ ($X = (\$183A)$) \rightsquigarrow X \rightarrow $\$183A$
(X points to memory spot $\$002C$)



EXTENDED

Same as DIRECT but with the argument being 16 bits long

INDEXED

Uses X register as the pointer to a given memory location

No offset: To address the memory spot pointed by X

8-bit offset: To address memory locations that are within 2^8 positions away from the one pointed by X

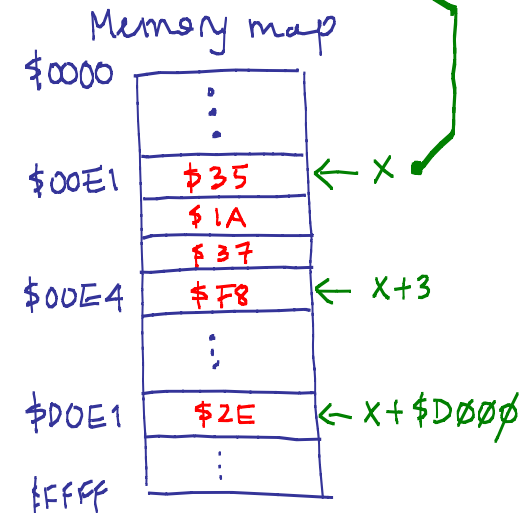
16-bit offset: To address memory locations that are within 2^{16} positions away from the one pointed by X

Assume X points to location $\$E1$

LDA $0, X$
(yields $A = \$35$)

LDA $3, X$
(yields $A = \$F8$)

LDA $\$D000, X$
(yields $A = \$2E$)



BUILDING PROJECT ϕ IN ASSEMBLY LANGUAGE

Brainstorming on some of the pieces...

③ Let's revisit the subroutine flow diagram

Note: The SWITCHES (1:4) are attached to PORTA (4:7)
The LEDS (1:4) are attached to PORTF (0:3)

STORING THE ACCUMULATOR IN MEMORY: STA

STA

Store Accumulator in Memory

STA

Operation

$M \leftarrow (A)$

Description

Stores the contents of A in memory. The contents of A remain unchanged. The N condition code is set if the most significant bit of A is set, the Z bit is set if A was \$00, and V is cleared. This allows conditional branching after the store without having to do a separate test or compare.

Condition Codes and Boolean Formulae

V		H		I	N	Z	C
0	1	1	—	—	1	1	—

V: 0
Cleared

N: A7
Set if MSB of result is 1; cleared otherwise

Z: $\overline{A7 \& A6 \& A5 \& A4 \& A3 \& A2 \& A1 \& A0}$
Set if result is \$00; cleared otherwise

Source Forms, Addressing Modes, Machine Code, Cycles, and Access Details

Source Form	Addr. Mode	Machine Code		HC-808 Cycles	Access Detail
		Opcode	Operand(s)		
STA opr8a	DIR	B7	dd	3	wpp
STA opr16a	EXT	C7	hh ll	4	pwpp
STA oprx16,X	IX2	D7	ee ff	4	pwpp
STA oprx8,X	IX1	E7	ff	3	wpp
STA ,X	IX	F7		2	wp
STA oprx16,SP	SP2	9ED7	ee ff	5	pwpp
STA oprx8,SP	SP1	9EE7	ff	4	pwpp

INCREMENTING A VALUE OR MEMORY LOCATION

INC

Increment

INC

Operation

$A \leftarrow (A) + \$01$
 Or $X \leftarrow (X) + \$01$
 Or $M \leftarrow (M) + \$01$

Description

Add 1 to the contents of A, X, or M. The V, N, and Z bits in the CCR are set or cleared according to the results of this operation. The C bit in the CCR is not affected; therefore, the BLS, BLO, BHS, and BHI branch instructions are not useful following an INC instruction.

INCX only affects the low-order byte of index register pair (H:X). To increment the full 16-bit index register pair (H:X), use AIX #1.

Condition Codes and Boolean Formulae

V	H	I	N	Z	C
1	1	1	—	—	—

V: $\overline{A7} \& R7$

Set if there was a two's complement overflow as a result of the operation; cleared otherwise. Two's complement overflow occurs if and only if (A), (X), or (M) was \$7F before the operation.

N: R7

Set if MSB of result is 1; cleared otherwise

Z: $R7 \& R6 \& R5 \& R4 \& R3 \& R2 \& R1 \& R0$

Set if result is \$00; cleared otherwise

Source Forms, Addressing Modes, Machine Code, Cycles, and Access Details

Source Form	Addr. Mode	Machine Code		HC808 Cycles	Access Detail
		Opocode	Operand(s)		
INC opr8a	DIR	3C	dd	5	rfwpp
INCA	INH (A)	4C		1	p
INCX	INH (X)	5C		1	p
INC opr8,X	IX1	6C	ff	5	rfwpp
INC ,X	IX	7C		4	rfwp
INC opr8,SP	SP1	9E5C	ff	6	prfwpp

INCX is recognized by assemblers as being equivalent to INCX.

