LDAA — Load Accumulator A — LDAA

Operation: (M) ⇒ A

Description: Loads the content of memory location M into accumulator A. The condition codes are set according to the data.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	0	–

N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: 0; cleared

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
LDAA #*opr8i*	IMM	86 ii	P	P
LDAA *opr8a*	DIR	96 dd	rPf	rfP
LDAA *opr16a*	EXT	B6 hh ll	rPO	rOP
LDAA *oprx0_xysp*	IDX	A6 xb	rPf	rfP
LDAA *oprx9,xysp*	IDX1	A6 xb ff	rPO	rPO
LDAA *oprx16,xysp*	IDX2	A6 xb ee ff	frPP	frPP
LDAA [D,*xysp*]	[D,IDX]	A6 xb	fIfrPf	fIfrfP
LDAA [*oprx16,xysp*]	[IDX2]	A6 xb ee ff	fIPrPf	fIPrfP

STAA — Store Accumulator A — STAA

Operation: (A) ⇒ M

Description: Stores the content of accumulator A in memory location M. The content of A is unchanged.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	0	–

N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: 0; cleared

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
STAA *opr8a*	DIR	5A dd	Pw	Pw
STAA *opr16a*	EXT	7A hh ll	PwO	wOP
STAA *oprx0_xysp*	IDX	6A xb	Pw	Pw
STAA *oprx9,xysp*	IDX1	6A xb ff	PwO	PwO
STAA *oprx16,xysp*	IDX2	6A xb ee ff	PwP	PwP
STAA [D,*xysp*]	[D,IDX]	6A xb	PIfw	PIfPw
STAA [*oprx16,xysp*]	[IDX2]	6A xb ee ff	PIPw	PIPPw

STAB — Store Accumulator B — STAB

Operation: (B) ⇒ M

Description: Stores the content of accumulator B in memory location M. The content of B is unchanged.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	0	–

N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: 0; cleared

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
STAB *opr8a*	DIR	5B dd	Pw	Pw
STAB *opr16a*	EXT	7B hh ll	PwO	wOP
STAB *oprx0_xysp*	IDX	6B xb	Pw	Pw
STAB *oprx9,xysp*	IDX1	6B xb ff	PwO	PwO
STAB *oprx16,xysp*	IDX2	6B xb ee ff	PwP	PwP
STAB [D,*xysp*]	[D,IDX]	6B xb	PIfw	PIfPw
STAB [*oprx16,xysp*]	[IDX2]	6B xb ee ff	PIPw	PIPPw

ADDA — Add without Carry to A — ADDA

Operation: (A) + (M) ⇒ A

Description: Adds the content of memory location M to accumulator A and places the result in A. This instruction affects the H status bit, so it is suitable for use in BCD arithmetic operations. See **DAA** instruction for additional information.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	Δ	–	Δ	Δ	Δ	Δ

H: $A3 \bullet M3 + M3 \bullet \overline{R3} + \overline{R3} \bullet A3$
Set if there was a carry from bit 3; cleared otherwise
N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: $A7 \bullet M7 \bullet \overline{R7} + \overline{A7} \bullet \overline{M7} \bullet R7$
Set if two's complement overflow resulted from the operation; cleared otherwise
C: $A7 \bullet M7 + M7 \bullet \overline{R7} + \overline{R7} \bullet A7$
Set if there was a carry from the MSB of the result; cleared otherwise

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
ADDA #*opr8i*	IMM	8B ii	P	P
ADDA *opr8a*	DIR	9B dd	rPf	rfP
ADDA *opr16a*	EXT	BB hh ll	rPO	rOP
ADDA *oprx0_xysp*	IDX	AB xb	rPf	rfP
ADDA *oprx9,xysp*	IDX1	AB xb ff	rPO	rPO
ADDA *oprx16,xysp*	IDX2	AB xb ee ff	frPP	frPP
ADDA [D,*xysp*]	[D,IDX]	AB xb	fIfrPf	fIfrfP
ADDA [*oprx16,xysp*]	[IDX2]	AB xb ee ff	fIPrPf	fIPrfP

INCA — Increment A — INCA

Operation: (A) + $01 ⇒ A

Description: Add one to the content of accumulator A.

The N, Z, and V status bits are set or cleared according to the results of the operation. The C status bit is not affected by the operation, thus allowing the INC instruction to be used as a loop counter in multiple-precision computations.

When operating on unsigned values, only BEQ, BNE, LBEQ, and LBNE branches can be expected to perform consistently. When operating on two's complement values, all signed branches are available.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	Δ	–

N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: Set if there is a two's complement overflow as a result of the operation; cleared otherwise. Two's complement overflow occurs if and only if (A) was $7F before the operation.

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
INCA	INH	42	O	O

CLRB — Clear B — CLRB

Operation: 0 ⇒ B

Description: All bits in accumulator B are cleared to 0.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	0	1	0	0

N: 0; cleared
Z: 1; set
V: 0; cleared
C: 0; cleared

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
CLRB	INH	C7	O	O

DECB — Decrement B — DECB

Operation: (B) – $01 ⇒ B

Description: Subtract one from the content of accumulator B.

The N, Z, and V status bits are set or cleared according to the results of the operation. The C status bit is not affected by the operation, thus allowing the DEC instruction to be used as a loop counter in multiple-precision computations.

CCR Details:

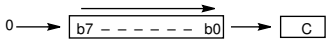
S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	Δ	–

N: Set if MSB of result is set; cleared otherwise
Z: Set if result is $00; cleared otherwise
V: 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 (B) was $80 before the operation.

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
DECB	INH	53	O	O

LSRA — Logical Shift Right A — LSRA

Operation:



Description: Shifts all bits of accumulator A one place to the right. Bit 7 is loaded with 0. The C status bit is loaded from the least significant bit of A.

CCR Details:

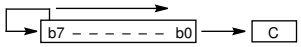
S	X	H	I	N	Z	V	C
–	–	–	–	0	Δ	Δ	Δ

N: 0; cleared
Z: Set if result is $00; cleared otherwise
V: $N \oplus C = [N \bullet \overline{C}] + [\overline{N} \bullet C]$ (for N and C after the shift)
Set if (N is set and C is cleared) or (N is cleared and C is set); cleared otherwise (for values of N and C after the shift)
C: A0
Set if the LSB of A was set before the shift; cleared otherwise

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
LSRA	INH	44	O	O

ASRA — Arithmetic Shift Right A — ASRA

Operation:



Description: Shifts all bits of accumulator A one place to the right. Bit 7 is held constant. Bit 0 is loaded into the C status bit. This operation effectively divides a two's complement value by two without changing its sign. The carry bit can be used to round the result.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	–	Δ	Δ	Δ	Δ

N: Set if MSB of result is set; cleared otherwise

Z: Set if result is $00; cleared otherwise

V: $N \oplus C = [N \bullet \overline{C}] + [\overline{N} \bullet C]$ (for N and C after the shift)
Set if (N is set and C is cleared) or (N is cleared and C is set); cleared otherwise (for values of N and C after the shift)

C: A0
Set if the LSB of A was set before the shift; cleared otherwise

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
ASRA	INH	47	O	O

SWI — Software Interrupt — SWI

Operation:
(SP) – $0002 ⇒ SP; $RTN_H : RTN_L \Rightarrow (M_{(SP)} : M_{(SP+1)})$
(SP) – $0002 ⇒ SP; $Y_H : Y_L \Rightarrow (M_{(SP)} : M_{(SP+1)})$
(SP) – $0002 ⇒ SP; $X_H : X_L \Rightarrow (M_{(SP)} : M_{(SP+1)})$
(SP) – $0002 ⇒ SP; B : A⇒ $(M_{(SP)} : M_{(SP+1)})$
(SP) – $0001 ⇒ SP; CCR ⇒ $(M_{(SP)})$
1 ⇒ I
(SWI Vector) ⇒ PC

Description: Causes an interrupt without an external interrupt service request. Uses the address of the next instruction after SWI as a return address. Stacks the return address, index registers Y and X, accumulators B and A, and the CCR, decrementing the SP before each item is stacked. The I mask bit is then set, the PC is loaded with the SWI vector, and instruction execution resumes at that location. SWI is not affected by the I mask bit. Refer to **Section 7. Exception Processing** for more information.

CCR Details:

S	X	H	I	N	Z	V	C
–	–	–	1	–	–	–	–

I: 1; set

Source Form	Address Mode	Object Code	Access Detail HCS12	Access Detail M68HC12
SWI	INH	3F	VSPSSPSsP(1)	VSPSSPSsP(1)

1. The CPU also uses the SWI processing sequence for hardware interrupts and unimplemented opcode traps. A variation of the sequence (VfPPP) is used for resets.