

- MC9S12 Assembler Directives
- A Summary of MC9S12 Instructions
- Disassembly of MC9S12 op codes
 - Review of Addressing Modes
 - Which branch instruction to use (signed vs unsigned)
 - Using X and Y registers as pointers
 - o Hand assembling a program
 - o How long does a program take to run?
 - Assembler directives
 - How to disassemble an MC9S12 instruction sequence

Summary of HCS12 addressing modes

ADDRESSING MODES

Na	me	Example	Op Code	Effective Address
INH	Inherent	ABA	18 06	None
IMM	Immediate	LDAA #\$35	86 35	PC + 1
DIR	Direct	LDAA \$35	96 35	0x0035
EXT	Extended	LDAA \$2035	B6 20 35	0x2035
IDX IDX1 IDX2	Indexed	LDAA 3,X LDAA 30,X LDAA 300,X	A6 03 A6 E0 13 A6 E2 01 2C	X + 3 X + 30 X + 300
IDX	Indexed Postincrement	LDAA 3,X+	A6 32	x (x+3 -> x)
IDX	Indexed Preincrement	LDAA 3,+X	A6 22	X+3 (X+3 -> X)
IDX	Indexed Postdecrement	LDAA 3,X-	A6 3D	x (x-3 -> x)
IDX	Indexed Predecrement	LDAA 3,-X	A6 2D	X-3 (X-3 -> X)
REL	Relative	BRA \$1050 LBRA \$1F00	20 23 18 20 0E CF	PC + 2 + Offset PC + 4 + Offset



A few instructions have two effective addresses:

• **MOVB #\$AA,\$1C00** Move byte 0xAA (IMM) to address

\$1C00 (EXT)

• **MOVW 0,X,0,Y** Move word from address pointed to by

X (IDX) to address pointed to by Y

(IDX)

A few instructions have three effective addresses:

• **BRSET FOO,#\$03,LABEL** Branch to LABEL (REL) if bits #\$03 (IMM) of variable FOO (EXT) are set.



Using X and Y as Pointers

- Registers X and Y are often used to point to data.
- To initialize pointer use

ldx #table

not

ldx table

• For example, the following loads the address of table (\$1000) into X; i.e., X will point to table:

ldx #table ; Address of table $\Rightarrow X$

The following puts the first two bytes of table (\$0C7A) into X. X will **not** point to table:

ldx table ; *First two bytes of table* $\Rightarrow X$

• To step through table, need to increment pointer after use

ldaa 0,x inx

or

ldaa 1,x+



table

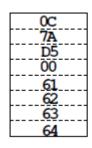


table: dc.b 12,122,-43,0

Which branch instruction should you use?

Branch if A > BIs 0xFF > 0x00?

If unsigned, 0xFF = 255 and 0x00 = 0, so 0xFF > 0x00

If signed, 0xFF = -1 and 0x00 = 0, so 0xFF < 0x00

Using unsigned numbers: **BHI** (checks C bit of CCR)

Using signed numbers: **BGT** (checks V bit of CCR)

For unsigned numbers, use branch instructions which check C bit

For signed numbers, use branch instructions which check V bit



Hand Assembling a Program

To hand-assemble a program, do the following:

- **1**. Start with the org statement, which shows where the first byte of the program will go into memory.
- (e.g., **org** \$2000 will put the first instruction at address \$2000.)
- **2**. Look at the first instruction. Determine the addressing mode used.
- (e.g., **ldab** #10 uses IMM mode.)
- **3**. Look up the instruction in the **MC9S12 S12CPUV2 Reference Manual**, find the appropriate Addressing Mode, and the Object Code for that addressing mode. (e.g., **ldab IMM** has object code **C6 ii**.)
 - Table A.1 of the S12CPUV2 Reference Manual has a concise summary of the instructions, addressing modes, op-codes, and cycles.
- **4**. Put in the object code for the instruction, and put in the appropriate operand. Be careful to convert decimal operands to hex operands if necessary. (e.g., **ldab** #10 becomes **C6** 0**A**.)
- **5**. Add the number of bytes of this instruction to the address of the instruction to determine the address of the next instruction. (e.g., \$2000 + 2 = \$2002 will be the starting address of the next instruction.)



org \$2000 ldab #10 loop: clra

dbne b,loop

swi

Freescale HC12-Assembler (c) Copyright Freescale 1987-2010

Abs. Rel. Loc Obj. code Source line

1 1 2 2 0000 2000 prog: equ \$2000 3 3 org prog 4 a002000 C60A ldab #10 5 a002002 87 loop: clra 6 a002003 0431 FC dbne b,loop 7 a002006 3F swi

Table A-1. Instruction Set Summary (Sheet 7 of 14)

	1	4.44	Markins	Arres	es Dotail		
Source Form	Operation	Addr. Mode	Machine Coding (hex)	HCS12	M68HC12	SXHI	NZVC
LBGT rah 6	Long Branch if Greater Than (if $Z + (N \oplus V) = 0$) (signed)	REL	18 2E qq rr	OPPP/GPO ¹	OPPP/OPO ¹		
LBHI rolt6	Long Branch if Higher (if C + Z = 0) (unsigned)	REL	18 22 qq rr	0999/090 ¹	OPPP/OPO ¹		
LBHS raft 6	Long Branch if Higher or Same (if C = 0) (unsigned) same function as LBCC	REL	18 24 qq rr	0999/090 ¹	OPPP/OPO ¹		
LBLE rah 6	Long Branch if Less Than or Equal (if $Z + (N \oplus V) = 1$) (signed)	REL	18 2F qq rr	OPPP/OPO ¹	OPPP/OPO ¹		
LBLO re/h6	Long Branch if Lower (if C = 1) (unsigned) same function as LBCS	REL	18 25 qq rr	0979/090 ¹	0PPP/0P0 ¹		
LBLS rah 6	Long Branch if Lower or Same (if C + Z = 1) (unsigned)	REL	18 23 qq rr	OPPP/OPO ¹	OPPP/OPO ¹		
LBLT re/h6	Long Branch if Less Than (if N ⊕ V = 1) (signed)	REL	18 2D qq rr	OPPP/GPO ¹	OPPP/OPO ¹		
LBMI ral 16	Long Branch if Minus (if N = 1)	REL	18 2B qq rr	0999/090 ¹	OPPP/OPO ¹		
LBNE raft 6	Long Branch if Not Equal (if Z = 0)	REL	18 26 qq rr	0999/090 ¹	OPPP/OPO ¹		
LBPL raft 6	Long Branch if Plus (if N = 0)	REL	18 2A qq rr	OPPP/OPO ¹	OPPP/OPO ¹		
LBRA raft 6	Long Branch Always (f 1-1)	REL	18 20 qq rr	OPPP	OPPP		
LBRN ral 16	Long Branch Never (ff 1 = 0)	REL	18 21 qq rr	OPO	OPO		
LBVC raft 6	Long Branch if Overflow Bit Clear (if V=0)	REL	18 28 qq rr	0999/090 ¹	OPPP/OPO ¹		
LBVS reht6	Long Branch if Overflow Bit Set (if V = 1)	REL	18 29 qq rr	0999/090 ¹	OPPP/OPOl		
LDAA #opr8i LDAA qor8a LDAA qor0 8a LDAA qor00 xysp LDAA qor01 xysp LDAA qor016 xysp LDAA [0,ysp] LDAA [0,ysp] LDAA [0,px16 xysp]	(M) → A Load Accumulator A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX2]	86 11 96 dd 86 hh 11 A6 xb A6 xb ff A6 xb en ff A6 xb en ff A6 xb en ff	P rPf rPO rPf rPO frPP fifrPf fifrPf	7 rf7 r07 rf7 r90 fr97 f1Frf7		ΔΔ0-
LDAB #opr8i LDAB opr8i LDAB opr1 fit LDAB opr0 //ysp LDAB opr0//ysp LDAB opr0///ysp LDAB (pxyp) LDAB (pxyp) LDAB (pxyp) LDAB (pxyp) LDAB (pxyp)	(M) → B Load Accumulator B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX]	C6 11 D6 dd F6 hh 11 E6 xb E6 xb ff E6 xb ee ff E6 xb E6 xb ee ff	P rPf rPO rPf rPO frPP fifrPf fiPrPf	9 169 109 199 199 1199 11119 11119		ΔΔ0-
LDD #opr16i LDD opr16a LDD opr16a LDD opr16a LDD opr02,yysp LDD opr02,yysp LDD opr02,yysp LDD opr016,yysp LDD [0,xysp] LDD [0,xysp]	(M:M+1) → A:B Load Double Accumulator D (A:B)	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	CC jj kk DC dd FC hh 11 EC xb EC xb ff EC xb ee ff EC xb ee ff	PO RPF RPO RPF RPO FRPP FIFRPF FIFRPF	OP REP ROP REP RPO ERPP EIFREP EIFREP		ΔΔ0-

Note 1. OPPPIOPO indicates this instruction takes four cycles to refill the instruction queue if the branch is taken and three cycles if the branch is not taken.

Table A-1. Instruction Set Summary (Sheet 3 of 14)

				Laure De	-3		
Source Form	Operation	Addr. Mode	Machine Coding (hex)	Access De HCS12	M68HC12	SXHI	NZVC
BLS note	Branch if Lower or Same (if C + Z = 1) (unsigned)	REL	23 rr	ppp/pl	ppp/p ¹		
BLT rale	Branch if Less Than (if N ⊕ V = 1) (signed)	REL	2D rr	PPP/p1	PPP/P1		
BMI rals	Branch if Minus (if N = 1)	REL	2B rr	ppp/p ¹	PPP/p ¹		
BNE rol8	Branch if Not Equal (if Z = 0)	REL	26 rr	ppp/p ¹	PPP/P1		
BPL add	Branch if Plus (if N = 0)	REL	2A rr	ppp/p ¹	ppp/p ¹		
BRAzelB	Branch Always (if 1 = 1)	REL	20 rr	222	PPP		
BRCLR oprisa, mskil, ralis BRCLR oprisa, mskil, ralis BRCLR oprisi, xysp, mskil, ralis BRCLR oprisi, xysp, mskil, ralis BRCLR oprisi yysp, mskil, ralis	Branch if (M) • (mm) = 0 (if All Selected Bit(s) Clear)	DIR EXT IDX IDX1 IDX2	4F dd mm rr 1F hh 11 mm rr 0F xb mm rr 0F xb ff mm rr 0F xb ee ff mm rr	#899 #1999 #899 #1999 #1999	rPPP rEPPP rPPP rEEPPP frPEEPPP		
BRN rale	Branch Never (f 1 = 0)	REL	21 rr	P	P		
BRSET opril, makil, ralil BRSET oprilia, makil, ralil BRSET oprili) xysp, makil, ralil BRSET oprili) xysp, makil, ralil BRSET oprili) xysp, makil, ralil	Branch if (M) • (mm) = 0 (if All Selected Bit(s) Set)	DIR EXT IDX IDX1 IDX2	4E dd mm rr 1E hh 11 mm rr 0E xb mm rr 0E xb ff mm rr 0E xb ee ff mm rr	#999 #1999 #999 #1999 #1999	:777 :E777 :P77 :E775 :E77EF777		
BSET opr8, msk8 BSET opr16u, msk8 BSET opr00_xysp, msk8 BSET opr00_xysp, msk8 BSET opr016,xysp, msk8	(M) + (mm) → M Set Bit(s) in Memory	DIR EXT IDX IDX1 IDX2	4C dd mm 1C hh 11 mm 0C xb mm 0C xb ff mm 0C xb ee ff mm	rPw0 rPw0 rPw0 rPw0 frPwP0	rPOw rPPw rPow rPw9 ErPw0P		ΔΔ0-
BSR reds	(SP) = 2 → SP; RTN _E RTN _L → M _(SP) M _(SP+1) Subroutine address → PC Branch to Subroutine	REL	07 rr	SPPP	PPPS		
BVC red8	Branch if Overflow Bit Clear (if V = 0)	REL	28 rr	ppp/p ¹	PPP/P1		
BVS ral8	Branch if Overflow Bit Set (if V = 1)	REL	29 rr	ppp/p ¹	PPP/p ¹		
CALL opri6a, page CALL opri0, xysp, page CALL opri0, xysp, page CALL opri0, xysp, page CALL [0, xysp, CALL [0, xysp] CALL [opri16, xysp]	(SP) − 2 → SP; RTN _L ATN _L → M _(SP) M _(SP+1) (SP) − 1 → SP; (PPG) → M _(SP) ; pg → PPAGE register; Program address → PC Call subroutine in extanded memory (Program may be located on another expansion memory page.) Indirect modes get program address and new pg value based on pointer.	EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	4A hh 11 pg 4B xb pg 4B xb ff pg 4B xb ee ff pg 4B xb ee ff pg 4B xb ee ff	gnSeppp gnSeppp gnSeppp fgnSeppp flignSeppp flignSeppp	gnfSaPPP gnfSaPPP gnfSaPPP fgnfSaPPP flignSaPPP flignSaPPP		
CBA	(A) – (B) Compare 8-Bit Accumulators	INH	18 17	00	00	-	ΔΔΔΔ
CLC	0 → C Translates to ANDCC #\$FE	IMM	10 FE	P	P		0
CII	0 → 1 Translates to ANDCC #\$EF (enables I-bit interrupts)	IMM	10 EF	P	P	0	
CLR opriša CLR opračujep CLR opračujep CLR (Dujep) CLR (Dujep) CLR (Sujep) CLR (Sujep) CLRA CLIB CLV	0 → M Clear Memory Location 0 → A Clear Accumulator A 0 → B Clear Accumulator B 0 → V	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	79 hh 11 69 xb 69 xb ff 69 xb ee ff 69 xb ee ff 69 xb ee ff 87 C7	PwO Pw PwO PwP Pifw Pifw PiPw O O	WOP PWO PWP PIEPW PIPPW O O		0-
	Translates to ANDCC #\$FD						

Note 1. PPP/P indicates this instruction takes three cycles to refill the instruction queue if the branch is taken and one program fetch cycle if the branch is not taken.

Table A-1. Instruction Set Summary (Sheet 4 of 14)

Source Form	Operation	Addr. Mode	Machine Coding (hex)	Access Detail HCS12	M68HC12	SXHI	NZVC
CMPB #opr8i	(B) - (M)	IMM	C1 11	p	P		ΔΔΔΔ
CMPB oprise	Compare Accumulator B with Memory	DIR	D1 dd	rPf	rfr		
CMPB opri6a	,	EXT	F1 hh 11	r90	ros		
CMPB opni0_xysp		IDX	El xb	rPf	rfF		
CMPB opnx8.xysp CMPB opnx16.xysp		IDX1 IDX2	El xb ff El xb ee ff	rPO frPP	rgo frgg		
CMPB (D,xysp)		ID.IDXI	El xb	fifepf	EIErER		
CMPB [qprx:16,xysp]		[IDX2]	El xb ee ff	fiprpf	EIPrEP		
COM opr16a	$(\overline{M}) \rightarrow M$ equivalent to SFF $-(M) \rightarrow M$	EXT	71 hh 11	rPw0	rOPw		ΔΔ01
COM aprix0_xysp	(si) → maquetation for arr = (si) → m 1's Complement Memory Location	IDX	61 xb	r?w	rPw		
COM oprast xysp	To a surprise to the surprise	IDX1	61 xb ff	r9w0	rPOw		
COM oproc16,xysp COM (D,xysp)		IDX2 ID,IDX1	61 xb ee ff 61 xb	frPeP fifrPe	frFFw fifrFw		
COM [aprox16,xysp]	The state of the s	10002	61 xb se ff	firre	EIPrPw		
COMA	(A) → A Complement Accumulator A	INH	41	0	0		
COMB	(B) → B Complement Accumulator B	INH	51	0	0		
CPD #aprt6i CPD apr8a	(A:B) - (M:M+1) Compare D to Membry (16-Bit)	DIR	8C jj kk 9C dd	PO RPE	OF REP		$\Delta\Delta\Delta\Delta\Delta$
CPD cont Sa	Compare D to seembly (16-bit)	EXT	BC hh 11	RPO	ROP		
CPD opnio_xysp		IDX	AC xb	RPE	REP		
CPD opnosuysp		IDX1	AC xb ff	RPO	RPO		
CPD quox16,xysp		ID002	AC xb ee ff	ERPP	free		
CPD [D,xysp] CPD [optx18,xysp]		[D,IDX] [DX2]	AC xb AC xb se ff	fifRPf fipRPf	FIFRER		
CPS#apr16i	(SP) - (M:M+1)	IMM	8F 11 kk	PO	OF		ΔΔΔΔ
CPS oprise	Compare SP to Memory (16-Bit)	DIR	9F dd	RPE	REP		
CPS opr16a		EXT	BF hh 11	RPO	ROP		
CPS oprx0_xysp		IDX IDX1	AF xb AF xb ff	RPE RPO	REP		
CPS opnx8,xysp CPS opnx18,xysp		IDX1	AF XD II	ERPP	ERFF		
CPS [D,xysp]		ID,IDXI	AF xb	fifapf	EIERER		
CPS [qp/xr16[xysp]		[1002]	AF xb ee ff	fipppf	EIPREP		
CPX #opr16i	(X) - (M:M+1)	IMM	BE jj kk	PO	OF		$\Delta\Delta\Delta\Delta\Delta$
CPX oprSu	Compare X to Memory (16-Bit)	DIR	9E dd	RPE	REP		
CPX opri6a CPX opri0_xysp		EXT	BE hh 11 AE xb	RPO RPE	ROF		
CPX opnx0.xysp		IDX1	AE xb ff	RPO	RPO		
CPX oprx16,xysp		IDX2	AE xb ee ff	ERPP	ferr		
CPX [D,xysp]		[D,IDX]	AE xb	EIERPE	fifzfr		
CPX [aprox16,xysp]		[IDX2]	AE xb ee ff	EIPRPE	fipsfp		
CPY #opr16i CPY opr8e	(Y) - (M:M+1) Compare Y to Memory (16-Bit)	DIR	8D jj kk 9D dd	PO RPE	OF REF		ΔΔΔΔ
CPY oprisa	Company 1 to Maritaly (16-bit)	EXT	BD hh 11	RP1 RPO	REF		
CPY opriou xysp		IDX	AD xb	RPE	REP		
CPY oproxit xysp		IDX1	AD xb ff	RPO	RPO		
CPY oprx18,xysp		IDX2	AD xb ee ff	frpp fifrpf	FIFRER		
CPY [D,xysp] CPY [aprx:16,xysp]		[D,IDX] 11DX(21	AD xb AD xb ee ff	figget	FIRRER		
DAA	Adjust Sum to BCD	INH	18 07	ofo	o£o		ΔΔ?Δ
	Decimal Adjust Accumulator A	me:					
DBEQ abdrys, rolli	(ontr) – 1→ ontr if (ontr) = 0, then Branch	(9-bit)	04 1b rr	PPP (branch) PPO (no	222		
	else Continue to next instruction	(a only		branch)			
	Decrement Counter and Branch if = 0 (ontr = A, B, D, X, Y, or SP)						
DBNE abdxys, ral9	(ontr) - 1 → ontr	REL	04 1b rr	PPP (branch)	PPP		
, .	If (ontr) not = 0, then Branch;	(9-bit)		PPO (no			
	else Continue to next instruction			branch)			
	Decrement Counter and Branch if ≠ 0						
	(ontr = A, B, D, X, Y, or SP)						
			.	l			



DBNE

Decrement and Branch if Not Equal to Zero

DBNE

Operation $(counter) - 1 \Rightarrow counter$

If (counter) not = 0, then (PC) + $$0003 + rel \Rightarrow PC$

Subtracts one from the counter register A, B, D, X, Y, or SP. Branches to a relative destination if the counter register does not reach zero. Rel is a 9-bit two's complement offset for branching forward or backward in memory. Branching range is \$100 to \$0FF (-256 to +255) from the address following the last byte of object code in the instruction.

CCR Effects

SXHINZVC

Code and CPU Cycles

Source Form Add		Machine Code (Hex)	CPU Cycles
DBNE abdxysp, rei9	REL (9-bit)		PPP (branch) PPO (no branch)

Loop Primitive Postbyte (1b) Coding									
Source Form	Postbyte ¹	Object Code	Counter Register	Offset					
DBNE A, rel9 DBNE B, rel9 DBNE D, rel9 DBNE X, rel9 DBNE Y, rel9 DBNE SP, rel9	0010 X000 0010 X001 0010 X100 0010 X101 0010 X110 0010 X111	04 20 rr 04 21 rr 04 24 rr 04 25 rr 04 26 rr 04 27 rr	A B D X Y SP	Positive					
DBNE A, rel9 DBNE B, rel9 DBNE D, rel9 DBNE X, rel9 DBNE Y, rel9 DBNE SP, rel9	0011 X000 0011 X001 0011 X100 0011 X101 0011 X110 0011 X111	04 30 rr 04 31 rr 04 34 rr 04 35 rr 04 36 rr 04 37 rr	A B D X Y SP	Negative					

NOTES:

^{1.} Bits 7:6:5 select DBEQ or DBNE; bit 4 is the offset sign bit: bit 3 is not used; bits 2:1:0 select the counter register.



MC9S12 Cycles

- MC9S12 works on 48 MHz clock
- A processor cycle takes 2 clock cycles P clock is 24 MHz
- Each processor cycle takes **41.7 ns** (1/24 μs) to execute
- An instruction takes from 1 to 12 processor cycles to execute
- You can determine how many cycles an instruction takes by looking up the CPU cycles for that instruction in the Reference Manual.
 - For example, **LDAB** using the **IMM** addressing mode shows one CPU cycle (of type P).
 - − **LDAB** using the **EXT** addressing mode shows three CPU cycles (of type **rPO**).
 - Section 6.6 of the S12CPUV2 Reference Manual explains what the HCS12 is doing during each of the different types of CPU cycles.

2000	org \$2000	; Inst	Mode	Cycles
2000 C6 0A	ldab #10	; LDAB	(IMM)	1
2002 87	loop:clra	; CLRA	(INH)	1
2003 04 31 FC	dbne b,loo	op; DBN	E (REL) 3
2006 3F	swi	; SWI		9



The program executes the **ldab** #10 instruction once. It then goes through the loop 10 times (which has two instructions, one with one cycle and one with three cycles), and finishes with the swi instruction (which takes 9 cycles).

Total number of cycles:

$$1 + 10 \times (1 + 3) + 9 = 50$$

$$50 \text{ cycles} = 50 \times 41.7 \text{ ns/cycle} = 2.08 \text{ μs}$$

LDAB

Load B

LDAB

Operation $(M) \Rightarrow B$

OF

 $imm \Rightarrow B$

Loads B with either the value in M or an immediate value.

CCR

Effects

5	Х	н		N	_	V	C
-	-	-	-	Δ	Δ	0	-

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

Z: Set if result is \$00; cleared otherwise V: Cleared

Code and CPU Cycles

Source Form	Address Mode	Machine Code (Hex)	CPU Cycles
LDAB #opr8i LDAB opr16a LDAB opr16a LDAB oprx0_xysppc LDAB oprx16,xysppc LDAB oprx16,xysppc LDAB [0,xysppc] LDAB [oprx16,xysppc]	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C6 ii D6 dd P6 hh 11 E6 xb E6 xb ff E6 xb ee ff E6 xb ee ff E6 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIFrPf



Assembler Directives

- In order to write an assembly language program it is necessary to use assembler **directives**.
- T hese are not instructions which the HC12 executes but are directives to the assembler program about such things as where to put code and data into memory.
- CodeWarrior has a large number of assembler directives, which can be found in the CodeWarrior help section.
- We will use only a few of these directives. (Note: In the following table, [] means an optional argument.) Here are the ones we will need:

Directive	Description		Example
Name		-	100
equ	Give a value to a symbol	len:	equ 100
org	Set starting value of location		org \$1000
	counter where code or data		
	will go		
dc.b	Allocate and initialize storage	var:	dc.b 2,18
	for 8-bit variables.	name	: dc.b "Jane"
	Place the bytes in successive		
	memory locations		
dc.w	Allocate and initialize storage	var:	dc.w \$ABCD
	for 16-bit variables.		
	Place the bytes in successive		
	memory locations		
ds.b	Allocate specified number of	Table	: ds.b 10
	8-bit storage places		
ds.w	Allocate specified number of	table:	ds.w 50
	16-bit storage spaces		
	3 1		
dcb.b	Fill memory with a given	init d	ata: dcb.b 100,0
	value:	_	·
	The first value is the number		
	of bytes to fill.		
	The second number is the		
	value to put into memory		



Using labels in assembly programs

A **label** is defined by a name followed by a colon as the first thing on a line. When the label is referred to in the program, it has the numerical value of the location counter when the label was defined.

Here is a code fragment using labels and the assembler directives dc and ds:

org \$2000

table1: dc.b \$23,\$17,\$f2,\$a3,\$56

table2: ds.b 5

var: dc.w \$43af

The CodeWarrior assembler produces a listing file (**.lst**). Here is the listing file from the assembler:

Freescale HC12-Assembler (c) Copyright Freescale 1987-2009

	Rel. Loc Obj. code	Source line		
1	1		org	\$2000
2	2 a002000 2317 F2A3	table1:	dc.b	\$23,\$17,\$f2,\$a3,\$56
	002004 56			
3	3 a002005	table2:	ds.b	5
4	4 a00200A 43AF	var:	dc.w	\$43af
5	5			

Note that **table1** is a name with the value of \$2000, the value of the location counter defined in the **org** directive. Five bytes of data



are defined by the **dc.b** directive, so the location counter is increased from \$2000 to \$2005.

Note that **table2** is a name with the value of \$2005. Five bytes of data are set aside for table2 by the **ds.b** 5 directive. The as12 assembler initialized these five bytes of data to all zeros. **var** is a name with the value of \$200a, the first location after table2.



HC12 Instructions

- 1. Data Transfer and Manipulation Instructions instructions which move and manipulate data (S12CPUV2 Reference Manual, Sections 5.3, 5.4, and 5.5).
- Load and Store load copy of memory contents into a register; store copy of register contents into memory.

LDAA \$2000 ; Copy contents of addr \$2000 into A STD 0,X ; Copy contents of D to addrs X and X+1

• Transfer — copy contents of one register to another.

TBA; Copy B to A

TFR X,Y; Copy X to Y

• Exhange — exchange contents of two registers.

XGDX ; Exchange contents of D and XEXG A,B ; Exchange contents of A and B

• Move — copy contents of one memory location to another.

MOVB \$2000,\$20A0; Copy byte at \$2000 to \$20A0

MOVW 2,X+,2,Y+ ; Copy two bytes from address held

; in X to address held in Y

; Add 2 to X and Y

2. Arithmetic Instructions — addition, subtraction, multiplication, division (**S12CPUV2 Reference Manual**, Sections 5.6, 5.8 and 5.12).

ABA ; Add B to A; results in A

SUBD \$20A1 ; Subtract contents of \$20A1 from D



INX ; Increment X by 1

MUL ; Multiply A by B; results in D

3. Logic and Bit Instructions — perform logical operations (**S12CPUV2 Reference Manual**, Sections 5.9, 5.10, 5.11, 5.13 and 5.14).

• Logic Instructions

ANDA \$2000 ; Logical AND of A with contents of

; \$2000

EORB 2,X ; Exclusive OR B with contents of

; address (X+2)

• Clear, Complement and Negate Instructions

NEG -2,X; Negate (2's comp) contents of

; address (X-2)

CLRA ; Clear ACC A

• Bit manipulate and test instructions — work with bits of a register or memory.

BITA #\$08 ; Check to see if Bit 3 of A is set BSET \$0002,#\$18 ; Set bits 3 and 4 of address \$0002

• Shift and rotate instructions

LSLA ; Logical shift left A

ASR \$1000 ; Arithmetic shift right value at address

; \$1000



4. Compare and test instructions — test contents of a register or memory (to see if zero, negative, etc.), or compare contents of a register to memory (to see if bigger than, etc.) (**S12CPUV2 Reference Manual**, Section 5.9).

TSTA ; (A)-0 -- set flags accordingly

CPX #\$8000 ; (X) - \$8000 -- set flags accordingly

5. Jump and Branch Instructions — Change flow of program (e.g., goto, if-then-else, switch-case) (**S12CPUV2 Reference Manual**, Sections 5.19, 5.20 and 5.21).

JMP L1 ; Start executing code at address label

; L1

BEQ L2 ; If Z bit set, go to label L2

DBNE X,L3 ; Decrement X; if X not 0 then goto L3 BRCLR \$1A,#\$80,L4 ; If bit 7 of addr \$1A clear, go to

; label L4

JSR sub1 ; Jump to subroutine sub1 RTS ; Return from subroutine

6. Interrupt Instructions — Initiate or terminate an interrupt call (**S12CPUV2 Reference Manual**, Section 5.22).

• Interrupt instructions

SWI; Initiate software interrupt RTI; Return from interrupt



7. Index Manipulation Instructions — Put address into X, Y or SP, manipulate X, Y or SP (**S12CPUV2 Reference Manual**, Section 5.23).

ABX ; Add (B) to (X)

LEAX 5,Y ; Put address (Y) + 5 into X

8. Condition Code Instructions — change bits in Condition Code Register (**S12CPUV2 Reference Manual**, Section 5.26).

ANDCC #\$f0 ; Clear N, Z, C and V bits of CCR

SEV ; Set V bit of CCR

9. Stacking Instructions — push data onto and pull data off of stack (**S12CPUV2 Reference Manual**, Section 5.24).

PSHA ; Push contents of A onto stack

PULX ; Pull two top bytes of stack, put into X

10. Stop and Wait Instructions — put MC9S12 into low power mode (S12CPUV2 Reference Manual, Section 5.27).

STOP ; Put into lowest power mode

WAI; Put into low power mode until next interrupt

11. Null Instructions

NOP; No operation

BRN ; Branch never



12. Instructions we won't discuss or use — BCD arithmetic, fuzzy logic, minimum and maximum, multiply-accumulate, table interpolation (**S12CPUV2 Reference Manual**, Sections 5.7, 5.16, 5.17, and 5.18).

Disassembly of an HC12 Program

• It is sometimes useful to be able to convert *HC12 op codes* into *mnemonics*.

For example, consider the hex code:

ADDR	DATA							
1000 C	6 05 CE	20 00	E6 01	18 (06 04	35	$\mathbf{E}\mathbf{E}$	3F

- To determine the instructions, use Table A-2 of the HCS12 Core Users Guide.
 - If the first byte of the instruction is anything other than \$18, use Sheet 1 of Table A.2. From this table, determine the number of bytes of the instruction and the addressing mode.
 For example, \$C6 is a two-byte instruction, the mnemonic is LDAB, and it uses the IMM addressing mode. Thus, the two bytes C6 05 is the op code for the instruction LDAB #\$05.
 - If the first byte is **\$18**, use Sheet 2 of Table A.2, and do the same thing. For example, **18 06** is a two byte instruction, the mnemonic is **ABA**, and it uses the **INH** addressing mode, so



there is no operand. Thus, the two bytes **18 06** is the op code for the instruction **ABA**.

- Indexed addressing mode is fairly complicated to disassemble. You need to use Table A.3 to determine the operand. For example, the op code \$E6 indicates LDAB indexed, and may use two to four bytes (one to three bytes in addition to the op code). The postbyte 01 indicates that the operand is 0,1, which is 5-bit constant offset, which takes only one additional byte. All 5-bit constant offset, pre and post increment and decrement, and register offset instructions use one additional byte. All 9-bit constant offset instructions use two additional bytes, with the second byte holding 8 bits of the 9 bit offset. (The 9th bit is a direction bit, which is held in the first postbyte.) All 16-bit constant offset instructions use three postbytes, with the 2nd and 3rd holding the 16-bit unsigned offset.
- Transfer (**TFR**) and exchange (**EXG**) instructions all have the op code **\$B7**. Use Table A.5 to determine whether it is **TFR** or an **EXG**, and to determine which registers are being used. If the most significant bit of the postbyte is **0**, the instruction is a transfer instruction.
- Loop instructions (Decrement and Branch, Increment and Branch, and Test and Branch) all have the op code **\$04**. To determine which instruction the op code **\$04** implies, and whether the branch is <u>positive</u> (forward) or <u>negative</u> (backward), use Table A.6. For example, in the sequence **04 35 EE**, the 04 indicates a loop instruction. The 35 indicates it is a **DBNE X** instruction (decrement register X and branch if



result is not equal to zero), and the direction is backward (negative). The **EE** indicates a branch of -18 bytes.

_

• Use up all the bytes for one instruction, then go on to the next instruction.

C6 05 \Rightarrow LDAB #\$05 two-byte LDAB, IMM addressing mode

CE 20 00 \Rightarrow LDX #\$2000 three-byte LDX, IMM addressing mode

E6 01 \Rightarrow LDAB 1,X two to four-byte LDAB,

IDX addressing mode. Operand 01 => 1,X, a 5b constant offset which uses only one postbyte two-byte ABA, INH addressing mode

04 35 EE ⇒ **DBNE X,(-18)** three-byte loop instruction Postbyte 35 indicates DBNE X, negative **3F** ⇒ **SWI** one-byte SWI, INH addressing

mode



Table A-2. CPU12 Opcode Map (Sheet 1 of 2)

00 †5				40 1	50 1	60 3-6		80 1		A0 3-6		C0 1		E0 3-6	
BGND	ANDCC	BRA	PULX	NEGA	NEGB	NEG	NEG	SUBA	SUBA	SUBA	SUBA	SUBB	SUBB	SUBB	SUBB
IH 1	IM 2	RL 2		IH 1	IH 1	ID 2-4	EX 3		DI 2	ID 2-4	EX 3	IM 2			EX 3
01 5	1111	211	313		51 1	61 3-6	71 4	81 1	91 3		B1 3	C1 1	D1 3	E1 3-6	F1 3
MEM	EDIV	BRN	PULY	COMA	COMB	COM	COM	CMPA	CMPA	CMPA	CMPA	CMPB	CMPB	CMPB	CMPB
IH 1	IH 1	RL 2	IH 1		IH 1		EX 3		DI 2			IM 2	DI 2		EX 3
02 1	12 ‡1		32 3			62 3-6		82 1	92 3		B2 3			E2 3-6	F2 3
INY	MUL	BHI	PULA	INCA	INCB	INC	INC	SBCA	SBCA	SBCA	SBCA	SBCB	SBCB	SBCB	SBCB
IH 1	IH 1	RL 2	IH 1	IH 1	IH 1				DI 2	ID 2-4	EX 3	IM 2	DI 2		EX 3
03 1	13 3	23 3/1	33 3		53 1	63 3-6					B3 3	C3 2	D3 3	E3 3-6	F3 3
DEY	EMUL	BLS	PULB	DECA	DECB	DEC	DEC	SUBD	SUBD	SUBD	SUBD	ADDD	ADDD	ADDD	ADDD
IH 1	IH 1		IH 1		IH 1	ID 2-4					EX 3	IM 3	DI 2		EX 3
04 , 3	14 1 ORCC	24 3/1 BCC	34 2 PSHX	44 1 LSRA	54 1 LSRB	64 3-6 LSR	74 4 LSR	84 1 ANDA	94 3 ANDA	A4 3-6		C4 1	D4 3 ANDB	E4 3-6 ANDB	F4 3
loop										ANDA	ANDA EX 3	ANDB IM 2			ANDB EX 3
RL 3	IM 2	RL 2 25 3/1	IH 1 35 2	IH 1	IH 1 55 1	ID 2-4 65 3-6	EX 3		DI 2 95 3		B5 3	IM 2 C5 1		ID 2-4 E5 3-6	EA 3
JMP	JSR	BCS	PSHY	ROLA	ROLB	ROL	ROL	BITA	BITA	BITA	BITA	BITB	BITB	BITB	BITB
		RL 2				ID 2-4			DI 2					ID 2-4	
	16 4	26 3/1				66 3-6	78 4			A6 3-6				E6 3-6	F6 3
JMP	JSR	BNE	PSHA	RORA	RORB	ROR	ROR	LDAA	LDAA	LDAA	LDAA	LDAB	LDAB	LDAB	LDAB
		RL 2		IH 1		ID 2-4	EX 3					IM 2		ID 2-4	EX 3
		27 3/1				67 3-6	77 4			A7 1	B7 1	C7 1		E7 3-6	F7 3
BSR	JSR	BEQ	PSHB	ASRA	ASRB	ASR	ASR	CLRA	TSTA	NOP	TFR/EXG	CLRB	TSTB	TST	l'rst "
		RL 2			ı		EX 3		IH 1		IH 2	IH 1	IH 1		EX 3
1.08 1	118 -	1 28 3/1	38 3	48 1	58 1	68 3-6	78 4	88 1	98 3	A8 3-6	B8 3	C8 1	D8 3	F8 3-6	F8 3
08 1 INX	18 - Page 2	28 3/1 BVC	38 3 PULC	48 1 ASLA				88 1 EORA	98 3 EORA		B8 3 EORA		D8 3 EORB	E8 3-6 EORB	F8 3 EORB
	Page 2	BVC	PULC	ASLA	ASLB	ASL	ASL	EORA	EORA	EORA	EORA	EORB	EORB	EORB	
INX	Page 2		PULC IH 1	ASLA IH 1	ASLB	ASL ID 2-4	ASL EX 3	EORA	EORA DI 2	EORA	EORA EX 3	EORB IM 2	EORB DI 2	EORB	F8 3 EORB EX 3 F9 3
INX	Page 2	BVC RL 2	PULC IH 1	ASLA IH 1	ASLB IH 1	ASL ID 2-4	ASL EX 3	EORA IM 2	EORA DI 2	EORA ID 2-4	EORA EX 3	EORB IM 2	EORB DI 2	EORB ID 2-4	
INX IH 1	Page 2 19 2 LEAY	BVC RL 2 29 3/1 BVS	PULC IH 1 39 2 PSHC	ASLA IH 1 49 1 LSRD	ASLB IH 1 59 1 ASLD	ASL ID 2-4 69 ‡2-4 CLR	ASL EX 3 79 3 CLR	EORA IM 2 89 1 ADCA	EORA DI 2 99 3 ADCA	EORA ID 2-4 A9 3-6 ADCA	EORA EX 3 B9 3 ADCA	EORB IM 2 C9 1 ADCB	EORB DI 2 D9 3 ADCB	EORB ID 2-4 E9 3-6	EX 3 F9 3 ADCB
INX IH 1	Page 2 19 2 LEAY	BVC RL 2 29 3/1 BVS	PULC IH 1 39 2 PSHC IH 1	ASLA IH 1 49 1 LSRD IH 1	ASLB IH 1 59 1 ASLD IH 1	ASL ID 2-4 69 ‡2-4 CLR ID 2-4	ASL EX 3 79 3 CLR EX 3	EORA IM 2 89 1 ADCA IM 2	EORA DI 2 99 3 ADCA	EORA ID 2-4 A9 3-6 ADCA ID 2-4	EORA EX 3 B9 3 ADCA	EORB IM 2 C9 1 ADCB	EORB DI 2 D9 3 ADCB DI 2	EORB ID 2-4 E9 3-6 ADCB ID 2-4	EX 3 F9 3 ADCB
INX IH 1	Page 2 19 2 LEAY ID 2-4	BVC RL 2 29 3/1 BVS RL 2	PULC IH 1 39 2 PSHC IH 1	ASLA IH 1 49 1 LSRD IH 1	ASLB IH 1 59 1 ASLD IH 1	ASL ID 2-4 69 ‡2-4 CLR ID 2-4	ASL EX 3 79 3 CLR EX 3	EORA IM 2 89 1 ADCA IM 2	EORA DI 2 99 3 ADCA DI 2	EORA ID 2-4 A9 3-6 ADCA ID 2-4	EORA EX 3 B9 3 ADCA EX 3	EORB IM 2 C9 1 ADCB IM 2	EORB DI 2 D9 3 ADCB DI 2	EORB ID 2-4 E9 3-6 ADCB ID 2-4	EX 3 F9 3 ADCB EX 3
INX IH 1 09 1 DEX IH 1 0A ‡7	Page 2 19 2 LEAY ID 2-4 1A 2	BVC RL 2 29 3/1 BVS RL 2 2A 3/1	PULC IH 1 39 2 PSHC IH 1 3A 3	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4	ASL EX 3 79 3 CLR EX 3 7A 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA	EORA EX 3 B9 3 ADCA EX 3 BA 3	EORB IM 2 C9 1 ADCB IM 2 CA 1	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB	EORB ID 2-4 E9 3-6 ADCB ID 2-4 EA 3-6	EX 3 F9 3 ADCB EX 3 FA 3
INX IH 1 09 1 DEX IH 1 0A ‡7	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB	EORB ID 2-4 E9 3-8 ADCB ID 2-4 EA 3-8 ORAB ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2	EORB ID 2-4 E9 3-8 ADCB ID 2-4 EA 3-8 ORAB ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL	ASLB H 1 59 1 ASLD H 1 5A 2 STAA DI 2 5B 2 STAB	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4 6B ‡2-4 STAB	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB	DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB	EORB ID 2-4 E9 3-6 ADCB ID 2-4 EA 3-6 ORAB ID 2-4 EB 3-8	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 STAA DI 2 STAB DI 2 5C 2	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4 6B ‡2-4 STAB ID 2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3 7C 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2 8B 2 8B 2	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2	DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2	EORB ID 2-4 E9 3-8 ADCB ID 2-4 EA 3-8 ORAB ID 2-4 EB 3-8 ADDB ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 5B 2 STAB DI 2	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4 6B ‡2-4 STAB ID 2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA EX 3	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2	DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2	EORB ID 2-4 E9 3-8 ADCB ID 2-4 EA 3-8 ORAB ID 2-4 EB 3-8 ADDB ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 4-6 BSET ID 3-5	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C ‡+5 Wavr SP 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 5B 2 STAB DI 2 5C 2 STD DI 2	ASL ID 2-4 69 ‡2-4 CLR ID 2-4 6A ‡2-4 STAA ID 2-4 6B ‡2-4 STAB ID 2-4 6C ‡2-4 STD ID 2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3 7C 3 STD EX 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2 8B 2 CPD IM 3	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA EX 3 BC 3 CPD EX 3	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3	DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2	EORB ID 2-4 E9 3-6 ADCB ID 2-4 EA 3-6 ORAB ID 2-4 EB 3-6 ADDB ID 2-4 EC 3-6 LDD ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3
INX IH 1 09 1 DEX IH 1 08 †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6	Page 2 LEAY ID 2-4 1A 2 LEAX ID 2-4 1B 2 LEAS ID 2-4 1C 4 BSET EX 4 1D 4	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$+5 wavr SP 1 3D 5	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3 4D 4	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 5B 2 STAB DI 2 5C 2 STD DI 2 5D 2	ASL ID 2-4 69 #2-4 STAA ID 2-4 STAB ID 2-4 6C #2-4 STD ID 2-4 6D #2-4 6D #2-4	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3 7C 3 STD EX 3 7D 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2 8B 2 CPD IM 3 8D 2	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9D 3	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-8 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AC 3-6	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA EX 3 BC 3 CPD EX 3 BD 3	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CD 2	DI 2 DA 3 ORAB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3	EORB ID 2-4 E9 3-6 ADCB ID 2-4 EA 3-6 ORAB ID 2-4 EB 3-8 ADDB ID 2-4 EC 3-6 LDD ID 2-4 EC 3-6	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FD 3
INX IH 1 09 1 DEX IH 1 08 †8 RTC IH 1 08 †8 RTII IH 1 0C 4-6 BSET ID 3-5 0D 4-8 BCLR	Page 2 	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BLT	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\frac{1}{2}\$ FSHD IH 1 3C \$\frac{1}{2}\$ FSHD IT SPHD IT SPH	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-6 4C 4 BSET DI 3 4D 4 BCLR	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 5B 2 STAB DI 2 5C 2 STD DI 2 5C 2 STD DI 2 5C 2 STD 2	ASL ID 2-4 69 #2-4 CLR ID 2-4 6A #2-4 STAA ID 2-4 6B #2-4 STD ID 2-4 6C #2-4 STD ID 2-4 6D #2-4 STY	ASL EX 3 79 3 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3 7C 3 STD EX 3 7D 5 STY	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2 8C 2 CPD IM 3 8D 2 CPY	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9D 3 CPY	EORA ID 2-4 A9 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AD 3-8 CPY	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA EX 3 BC 2 CPD EX 3 BD 3 CPY	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CD 2 LDY	DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDD DI 2 DD 3	EORB ID 2-4 E9 3-6 ADCB ID 2-4 EA 3-6 ORAB ID 2-4 EB 3-6 ADDB ID 2-4 EC 3-6 LDD ID 2-4 ED 3-8 LDV	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FD 3 LDY
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 46 BSET ID 3-5 0D 4-6 BCLR ID 3-5	Page 2 LEAY ID 2-4 1A 2 LEAS ID 2-4 1B 2 LEAS ID 2-4 1C 4 BSET EX 4 1D 4 BSCT EX 4	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BLT RL 2	PULC H	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3 4D 4 BCLR DI 3	ASLB IH 1 59 1 ASLD IH 1 5A 2 STAA DI 2 5B 2 STAB DI 2 5C 2 STD DI 2 5C 2 STD DI 2 5TD 2 STY DI 2	ASL D 2-4 69	ASL EX 3 79 CLR EX 3 7A 3 STAA EX 3 7B 3 STAB EX 3 7C 3 STO 3 STO 3 STO 3 STY EX 3 STY EX 3	EORA IM 2 89 1 ADCA IM 2 8A 1 ORAA IM 2 8B 1 ADDA IM 2 8C 2 CPD IM 3 8D 2 CPY IM 3	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9D 3 CPY DI 2	EORA ID 2-4 A0 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AD 3-6 CPV ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA 3 ORAA EX 3 BB 3 ADDA EX 3 BC 3 CPD EX 3 BD 3 CPD EX 3 BD 3 CPY EX 3	EORB M 2 C9 1 ADCB M 2 CA 1 ORAB M 2 CB 1 ADDB M 2 CC 2 LDD M 3 CD 2 LDY M 3 CD 2 LDY M 3	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDD DI 2 DD 3 LDD	EORB ID 2-4 E9 3-8 ADCB ID 2-4 EA 3-6 ORAB ID 2-4 EB 3-6 ADDB ID 2-4 EC 3-6 LDD ID 2-4 ED 3-8 LDY ID 2-4	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FD 3 LDD 5 EX 3 FD 3 LDD 5 EX 3
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6 BCLR ID 3-5 0E ‡4-6	Page 2 LEAY ID 2-4 1A 2 LEAX ID 2-4 1B 2 LEAS ID 2-4 1C 4 BSET EX 4 1D 4 BCK BCK BCK BCK BCK BCK BCK BCK BCK BCK	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BGE RL 2 2D 3/1	PULC IH 1 29 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\pm\$+5 Wavr SP 1 3D 5 RTS IH 1 3E \$\pm\$17 THE PULL IN T	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B \$7-10 CALL ID 2-5 4C 4 BSET DI 3 4D 4 BCLR DI 3	ASLB H 1 59 1 ASLD 1 ASLD 1 ASLD 1 H 1 5 5 C STAA DI 2 5 STAB DI 2 5 C STD DI 2 5 STY DI 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 2 5 5 C 5 5 C 5 5 C 5 5	ASL D 2-4 69 ‡2-4 CLR D 2-4 6A ‡2-4 STAA D 2-4 6B \$2-4 STAB D 2-4 6C \$2-4 STD D 2-4 6D \$2-4 STY D D 2-4 6E \$2-4	ASL EX 3 79 CLR EX 3 7A 3 STAA EX 3 7B STAB EX 3 7C 3 STD EX 3 7D STAB	EORA M 2 89 1 ADCA M 2 8A 1 ORAA M 2 8B 1 ADDA M 2 8C 2 CPD M 3 8B 2 CPY M 3 8E 2 8 CP	EORA DI 2 99 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9C 7 9D 2 9C 9D 2 9C 9D 2 9E 3	EORA ID 2-4 AQ 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AD 3-6 CPY ID 2-4 AB 3-6 ADDA ADDA 3-6 ADDA ADDA 3-6	EORA EX 3 B9 3 ADCA S	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CE 2 C CD 2 LDY IM 3 CE 2 CE 2 CD 2 CD 2 CD 2 CD 2 CD 2 CD 2	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDD DI 2 DD 3 LDY DI 2 DE 3	EORB ID 24 E9 3-6 ADCB ID 24 EB 3-6 ADCB ID 24 EB 3-6 ADCB ID 24 EC 3-6 LDD ID 24 ED 3-6 LDY ID 24 EE 3-6 ED 4 ED 3-6 LDY ID 24 EE 3-6 ED 3-	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FD 3 LDY EX 3 FE 3
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6 BCLR ID 3-5 0E ‡4-6 BRSET	Page 2 LEAY ID 2-4 18 2 LEAS ID 2-4 18 2 LEAS ID 2-4 10 2-4 10 2-4 BSET BSET EX 4 10 4 BCLR EX 4 BCLR EX 4 10 5 BRSET	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI BLT RL 2 2C 3/1 BLT RL 2 2D 3/1 BLT RL 2 2D 3/1 BLT RL 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\frac{1}{2}\$ PSHD IH 1 3C \$\frac{1}{2}\$ PSHD IH 1 3C \$\frac{1}{2}\$ PSHD IH 1 3D 5 RTS IH 1 3E \$\frac{1}{2}\$ RTS WAIT	ASLA IH 1 1 LSRD IH 1 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C BSET DI 3 4D 4 BCLR DI 3 4E 4 BRSET	ASLB H 1 5 6 1 1 6 1 1 6 1 1 1	ASL 2-4 60 2-4 60 2-4 60 5TAA ID 2-4 65 5TAB ID 2-4 65 5TD ID 2-4 65 5TD ID 2-4 65 5TD ID 2-4 65 5TY ID 2-4 65 5TX	ASL EX 3 79 CLR EX 3 7A 3 STAA EX 3 STAB EX 3 7C TD EX 3 7D 3 STY EX 3 7TE 5 STY EX 3 STX	EORA M 2 89 1 ADCA M 2 8A 1 ORAA M 2 8B 1 ADDA M 2 8C CPD M 3 8D 2 CPY M 3 8E 2 CPX M 3 8E 2 CPX	EORA DI 2 99 3 ADCA DI 2 90 3 ADCA DI 2 9A 3 ADDA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9C 3 CPY DI 2 9E 3 CPX	EORA ID 2-4 A8 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 CPD ID 2-4 AC 3-6 CPY ID 2-4 AE 3-6 CPY ID 2-4 AE 3-6 CPX	EORA EX 3 B9 3 ADCA EX 3 BA 3 ADDA EX 3 BB 3 CPX EX 3 BD 3 CPD EX 3 BD 3 CPX EX 3 BB 3 CPX EX 3 BC CPD EX 3 BD 3 CPX EX 3 BC CPD EX 3 BC CPD EX 3 BC CPD EX 3 BC CPD EX 3 BC CPX EX 3 CPX EX 5 CPX EX 6 CPX EX 5 C	EORB M 2 C9 1 ADCB M 2 CA CA CA CA CA CA CA	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDY DI 2 DE 3 LDY DI 2 DE 3	EORB D 24 E9 3-6 ADCB D 24 EA 3-6 ORAB D 24 EB 3-6 ADDB D 24 EC 3-6 LDD D 24 ED 3-6 LDV D 24 EE 3-6 LDV D 24 EE 3-6 LDX EE 3-6 LDX D 24 EE 3-6 LDX D 3-6 LDX D 3-6	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FC 3 LDD EX 3 FC 3 LDD EX 3 FD 3 LDD EX 3 FD 3 LDD 3 EX 3 FD 3 LDD 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6 BCLR ID 3-5 0E ‡4-6 BRSET ID 4-6	Page 2	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BLT RL 2 2E 3/1 BLT RL 2	PULC IH 1 3 30 2 2 PSHC IH 1 3A 3 PULD IH 1 3C \$\frac{1}{2}\$ PSHD IH 1 3C \$\frac{1}{2}\$ PSHD IH 1 3C \$\frac{1}{2}\$ FS P 1 3D 5 RTS IH 1 3E \$\frac{1}{2}\$ RTS IH 1 1 3E \$\frac{1}{2}\$ TWAI IH 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3 4D 4 BCLR DI 3 4E 4 BCLR DI 3 4E 4 BCLR DI 3 EX 4 BCLR DI 3 DI 4 BCLR DI 3 DI 4 BCLR DI 3 DI 4 BCLR DI 3 DI 4 BCLR DI 3 DI 5 DI 5	ASLB IH 1 5 50 1 1 50 1 1 5 5 5 1 2 5 5 1 2 5 5 1 2 5 5 1 2 5 5 5 1 2 5 5 5 1 2 5 5 5 5	ASL D 2-4 68 ±2-4 CLR D 2-4 68 ±2-4 STAB D 2-4 66 ±2-4 STD D 2-4 66 ±2-4 STD D 2-4 66 ±2-4 STD D 2-4 66 ±2-4 STV D 2-4 66 ±2-4 STV D 2-4 67 57 C C C C C C C C C	ASL EX 3 79 3 CLR EX 3 7A 3 STAA 8 STAB EX 3 7C 3 STD EX 3 7D STD EX 3 7E 3 STX EX 3 STX STX EX 3 STX	EORA M 2 89 1 ADCA M 2 8 A 1 ORAA M 2 8 B 1 ADDA M 2 8 CPD M 3 8 D 2 CPV M 3 8 E 2 CPX M 3 8 E 2 CPX M 3 8 E 2 CPX M 3	EORA DI 2 90 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9C 3 CPD DI 2 9D 3 CPD DI 2 9E 3 CPY DI 2 9E 3 CPY DI 2	EORA ID 2-4 A9 3-6 ADCA ID 2-4 A8 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AE 3-6 CPV ID 2-	EORA EX 3 BP 3 ADCA EX 3 BA ORAA EX 3 BC CPD EX 3 BB 3 CPX EX 3 BC CX EX 3 EX 5 CX EX 3 EX 5 CX EX 3	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CD 2 LDV IM 3 CE 2 LDV IM 3 CE 2 LDV IM 3	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 DD 3 LDD DI 2 DD 3 DD 3 DD 3 DD 3 DD 3 DD 3 DD 1 DD 1	EORB D 24 E8 3-6 ADCB D 24 E8 3-6 ADCB D 24 E8 3-6 ADCB D 24 EC 3-6 LDD D 24 EE 3-6 LDV D 24 LD	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD 5 EX 3 FD 3 LDV EX 3 FD 3 LDV EX 3 FD 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6 BRSET ID 3-5 0E ‡4-6 BRSET ID 4-6 0F ‡4-6	Page 2 LEAY ID 2-4 1A 2 LEAX ID 2-4 1B 2 LEAS ID 2-4 1C 4 BSET EX 4 BCLR EX 4 1D 4 BCLR EX 5 BRET EX 5	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BGE RL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	PULC IH 1 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\frac{1}{3}\$ PS P 1 S PS P 1 3C \$\frac{1}{3}\$ PS P 1 3C \$\frac{1}{3}\$ PS P 1 3 S PS P 1 S P 1 S P 1 S P 1 S P 1 S P 1 S P 1 S P 1 S P 1	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3 4B CR DI 3 4B RSET DI 4 BRSET DI 3 4E 4 4 BRSET DI 4 BRSET D	ASLB IH 1 59 1 ASLD IH 1 1 50 2 STAA DI 2 STAB	ASL D 2-4 69	ASL EX 3 79 CLR EX 3 7A 3 STAB EX 3 7C 3 STAB EX 3 7D EX 3 7D EX 3 7E STAB EX 3 7D STY EX 3 7E STAB EX 3 7T S	EORA M 2 89 1 ADCA M 2 80 1 ADCA M 2 8 ADCA M 3	EORA DI 2 90 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9B 3 CPD DI 2 9C 3 CPY DI 2 9E 3 CPY DI 2 9E 3 CPY DI 2 9E 3	EORA ID 2-4 AR 3-6 ORAA ID 2-4 AR 3-6 ORAA ID 2-4 AR 3-6 CPU ID 2-4 AR 3-6 CPV ID 2-4 AR 3-6 CPV ID 2-4 AR 3-6 CPX ID 2-4 AR 3-6 3-6 CPX ID 2-4	EORA EX 3 B9 3 ADCA EX 3 BA ORAA EX 3 BB 3 ADDA EX 3 BC CPD EX 3 BB C CPY EX 3 BB C CPX EX 5 BF 3 CPX EX 5 BF 5 S BF 5 S S CPX EX 5 BF 5 S CPX EX 5 BP 5 S	EORB IM 2 C9 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CC LDX IM 3 CF 2 CF 2 CD 2 CD 3 CD 2 CD 3 CD 3 CF 2 CD 3	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DC 3 LD DI 3 LD DI 2 DC 3 LD DI 3 LD DI 3 LD DI 3 LD DI 5 LD DI 5 LD DI 5 LD DI 7 LD D D D D D D D D D D D D D D D D D D	EORB ID 24 EA 3-6 ORAB ID 24 EB 3-6 ADDB ID 24 EC 3-6 LDD ID 24 ED 3-6 LDD ID 24 EE 3-6 LDY ID 24 EE 3-6 LDY ID 24 EE 3-6 LDX ID 24 EF 3-6 EF 3-6	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 ADDB EX 3 ADDB EX 3 LDD EX 3 LDD EX 3 LDD EX 3 LDD EX 3 FB 3 ADDB EX 3 FB 3 ADDB EX 3 FB 3 ADDB EX 3 B EX 3 B EX 3 EX 3 B EX 3 EX 3 EX 3 EX 3
INX IH 1 09 1 DEX IH 1 DA ‡7 RTC IH 1 08 †8 RTI IH 1 0C 4-6 BSET ID 3-5 0D 4-6 BRSET ID 4-6 BRSE	Page 2	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGT RL 2 2D 3/1 BGT RL 2 25 3/1 BGT RL 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\frac{1}{3}\$ SP 1 1 3D 5 RTS IH 1 3E \$\frac{1}{3}\$ SW IH 1 3F 9 SW	ASLA IH 1 49 1 LSRD IH 1 48 \$7.10 CALL EX 4 48 \$7.10 CALL ID 2-5 4C 4 BSET DI 3 4D 4 BRSET DI	ASLB H 1 5 6 1 1 6 5 1 1 6 1 1 6	ASL 1D 2-4 69 ‡2-4 CLR 1D 2-4 6A ‡2-4 STAA 1D 2-4 6B ‡2-4 STD 1D 2-4 6D ‡2-4 STY 1D 2-4 6E ‡2-4 STX 1D 2-4 6F ‡2-4 STX 1D 2-4 6F ‡2-4 STS	ASL EX 3 779 3 CLR EX 3 77A 3 STAA 5 STAB EX 3 77C 3 STD EX 3 77D 3 STY EX 3 77E 3 STX EX 3 77F 3 STS STS STS STS STS STS STS STS STS S	EORA M 2 89 1 ADCA M 2 8A 1 ORAA M 2 8B 1 ADDA M 2 8C PD M 3 8D CPY M 3 8E CPX M 3 8F 2 CPS	EORA DI 2 99 3 ADCA DI 2 90 3 ADCA DI 2 9A 3 ADDA DI 2 9C 3 CPD DI 2 9C 3 CPV DI 2 9F 3 CPS	EORA ID 2-4 A8 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AD 3-6 CPY ID 2-4 AE 3-6 CPX ID 2-4 AF 3-6 CPS	EORA EX 3 B9 3 ADCA EX 3 BA AS ADDA EX 3 BB CPS CPD EX 3 BB CPS EX 3 BB CPS EX 3 BB CPS EX 3 BC CPD EX 3 BC CPD EX 3 BC CPD EX 3 BC CPD EX 3 BC CPS EX 3 BC CPS EX 3 BC CPS EX 3 BF 3 CPS EX 3 BF 3 CPS	EORB IM 2 CQ 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CD 2 LDY IM 3 CF 2 LDX IM 3 CF 2 LDX IM 3 CF 2 LDX	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDY DI 2 DE 3 LDX DI 2 DF 3 LDS	EORB D 24 E9 3-6 ADCB D 24 EA 3-6 ORAB D 24 EB 3-6 ADDB D 24 EC 3-6 LDY D 24 EC 3-6 LDX D 24 EF 3-6 LDX D 3-	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 FB 3 ADDB EX 3 FC 3 LDD EX 3 FD 3 LDY EX 3 FD 3 LDY EX 3 FD 3 FD 3 FD 3 LDY EX 3 FD 3 FD 3 FD 3 FD 3 FD 3 FD 3 LDY EX 3 FD 3 LDY EX 3 FD 3 LDY EX 3 FD 3 LDY EX 3 LDY EX 3 LDY EX 3 LDX EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX
INX IH 1 09 1 DEX IH 1 0A ‡7 RTC IH 1 0B 18 RTII IH 1 0C 4-6 BSET ID 3-5 0E ‡4-6 BRSET D 4-6 0F ‡4-6 0F ‡4-6	Page 2	BVC RL 2 29 3/1 BVS RL 2 2A 3/1 BPL RL 2 2B 3/1 BMI RL 2 2C 3/1 BGE RL 2 2D 3/1 BGE RL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	PULC IH 1 39 2 PSHC IH 1 3A 3 PULD IH 1 3B 2 PSHD IH 1 3C \$\frac{1}{3}\$ SP 1 1 3D 5 RTS IH 1 3E \$\frac{1}{3}\$ SW IH 1 3F 9 SW	ASLA IH 1 49 1 LSRD IH 1 4A ‡7 CALL EX 4 4B ‡7-10 CALL ID 2-5 4C 4 BSET DI 3 4B CR DI 3 4B RSET DI 4 BRSET DI 3 4E 4 4 BRSET DI 4 BRSET D	ASLB IH 1 59 1 ASLD IH 1 1 50 2 STAA DI 2 STAB	ASL D 2-4 69	ASL EX 3 779 3 CLR EX 3 77A 3 STAA 5 STAB EX 3 77C 3 STD EX 3 77D 3 STY EX 3 77E 3 STX EX 3 77F 3 STS STS STS STS STS STS STS STS STS S	EORA M 2 89 1 ADCA M 2 8A 1 ORAA M 2 8B 1 ADDA M 2 8C PD M 3 8D CPY M 3 8E CPX M 3 8F 2 CPS	EORA DI 2 90 3 ADCA DI 2 9A 3 ORAA DI 2 9B 3 ADDA DI 2 9B 3 CPD DI 2 9C 3 CPY DI 2 9E 3 CPY DI 2 9E 3 CPY DI 2 9E 3	EORA ID 2-4 A8 3-6 ADCA ID 2-4 AA 3-6 ORAA ID 2-4 AB 3-6 ADDA ID 2-4 AC 3-6 CPD ID 2-4 AD 3-6 CPY ID 2-4 AE 3-6 CPX ID 2-4 AF 3-6 CPS	EORA EX 3 B9 3 ADCA EX 3 BA 3 ADDA EX 3 BB 3 CPX EX 3 BB CPX EX 3 BB CPX EX 3 BB CPX EX 3 BC CPD EX 3 BC CPD EX 3 BC CPX EX 3 BF 3 CPX EX 3 BF 3 CPX	EORB IM 2 CQ 1 ADCB IM 2 CA 1 ORAB IM 2 CB 1 ADDB IM 2 CC 2 LDD IM 3 CD 2 LDY IM 3 CF 2 LDX IM 3 CF 2 LDX IM 3 CF 2 LDX	EORB DI 2 D9 3 ADCB DI 2 DA 3 ORAB DI 2 DB 3 ADDB DI 2 DC 3 LDD DI 2 DD 3 LDY DI 2 DE 3 LDX DI 2 DF 3 LDS	EORB ID 24 EA 3-6 ORAB ID 24 EB 3-6 ADDB ID 24 EC 3-6 LDD ID 24 ED 3-6 LDD ID 24 EE 3-6 LDY ID 24 EE 3-6 LDY ID 24 EE 3-6 LDX ID 24 EF 3-6 EF 3-6	EX 3 F9 3 ADCB EX 3 FA 3 ORAB EX 3 ADDB EX 3 FD 3 LDD EX 3 LDY EX 3 FC 3 LDY EX 3 FC 3 LDY EX 3 FC 3 LDY EX 3 FC 3 LDY EX 3 FC 3 LDY EX 3 LDX EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX 3 EX

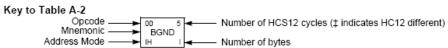


Table A-2. CPU12 Opcode Map (Sheet 2 of 2)

		20 4 LBRA	30 10 TRAP	40 10 TRAP	50 10 TRAP	60 10 TRAP	70 10 TRAP	80 10 TRAP	90 10 TRAP	A0 10 TRAP	B0 10 TRAP	C0 10 TRAP			F0 10
MOVW IM-ID 5	IDIV IH 2					IH 2							TRAP	TRAP	TRAP
IM-ID 5	11 12	21 3	IH 2 31 10			61 10	IH 2	IH 2 81 10	IH 2 91 10			IH 2	IH 2		F1 10
MOVW	FDIV	LBRN	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
EX-ID 5	IH 2	RL 4	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2		IH 2					
02 5	12 13	22 4/3	32 10	42 10		62 10		82 10			B2 10	C2 10	D2 10	E2 10	F2 10
MOVW ID-ID 4	EMACS SP 4	LBHI RL 4	TRAP	TRAP	TRAP	TRAP IH 2	TRAP IH 2	TRAP							
03 5	13 3	23 4/3			53 10			83 10					D3 10		F3 10
MOVW	EMULS	LBLS	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
IM-EX 6	IH 2	RL 4	IH 2	IH 2						IH 2	IH 2				IH 2
04 6 MOVW	14 12 EDIVS	24 4/3 LBCC	34 10 TRAP	44 10 TRAP	54 10 TRAP	64 10 TRAP	74 10 TRAP	84 10 TRAP	94 10 TRAP	A4 10 TRAP	B4 10 TRAP	C4 10 TRAP	D4 10 TRAP	E4 10 TRAP	F4 10 TRAP
EX-EX 6			IH 2	IH 2						IH 2	IH 2				IH 2
05 5	15 12	25 4/3			55 10	65 10	75 10		95 10		B5 10		D5 10		F5 10
MOVW	IDIVS	LBCS	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
ID-EX 5	IH 2		IH 2	IH 2	IH 2	IH 2	IH 2	IH 2		IH 2					
ABA	SBA	26 4/3 LBNE	36 10 TRAP	46 10 TRAP	56 10 TRAP	66 10 TRAP	76 10 TRAP	86 10 TRAP	96 10 TRAP	A6 10 TRAP	B6 10 TRAP	C6 10 TRAP	D6 10 TRAP	E6 10 TRAP	F6 10 TRAP
		RL 4		IH 2		IH 2	IH 2	IH 2		IH 2					
073	17 2	27 4/3				67 10	77 10	87 10			B710	C710	D7 10	E710	F710
DAA	CBA	LBEQ	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
IH 2	IH 2	RL 4 28 4/3	IH 2	IH 2 48 10	IH 2 58 10	IH 2	IH 2 78 10	IH 2 88 10	IH 2 98 10	IH 2 A8 10	IH 2 B8 10	IH 2 C8 10	IH 2 D8 10		IH 2 F8 10
MOVB	MAXA	LBVC	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
IM-ID 4	ID 3-5	RL 4	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2	IH 2
	19 4-7	29 4/3			59 10										F9 10
MOVB EX-ID 5	MINA ID 3-5	LBVS RL 4	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP	TRAP
0A 5		2A 4/3			5A 10										FA 10
MOVB	EMAXD	LBPL	REV	TRAP											
ID-ID 4		RL 4	SP 2			IH 2				IH 2					
MOVB 4	1B 4-7 EMIND	2B 4/3 LBMI	3B †5n/3n REVW	4B 10 TRAP	5B 10 TRAP	6B 10 TRAP	7B 10 TRAP	8B 10 TRAP	9B 10 TRAP	AB 10 TRAP	BB 10 TRAP	CB 10 TRAP	DB 10 TRAP	EB 10 TRAP	FB 10 TRAP
IM-EX 5	ID 3-5	RL 4			IH 2	IH 2	IH 2	IH 2		IH 2					
0C 6	1C 4-7		3C ‡†7B	4C 10	5C 10	6C 10	7C 10		9C 10				DC 10		FC 10
MOVB	MAXM	LBGE	WÁV	TRAP											
EX-EX 6	ID 3-5	RL 4	SP 2	IH 2	IH 2	IH 2	IH 2	IH 2		IH 2					
MOVB 5	1D D4-7 MINM	2D 4/3 LBLT	3D ‡6 TBL	4D 10 TRAP	5D 10 TRAP	6D 10 TRAP	TRAP	8D 10 TRAP	9D 10 TRAP	AD 10 TRAP	BD 10 TRAP	CD 10 TRAP	DD 10 TRAP	ED 10 TRAP	FD 10 TRAP
ID-EX 5	ID 3-5	RL 4				IH 2		IH 2		IH 2	IH 2	IH 2	IH 2		IH 2
	1E 4-7				5E 10	6E 10	7E 10	8E 10	9E 10	AE 10	BE 10	CE 10	DE 10	EE 10	FE 10
TAB	EMAXM	LBGT	STOP	TRAP											
IH 2	ID 3-5	RL 4	IH 2	IH 2 4F 10	IH 2 5F 10	6F 10	7F 10	IH 2 8F 10	9F 10	IH 2 AF 10	IH 2 BF 10	IH 2 CF 10	IH 2	IH 2	IH 2
TBA	EMINM	LBLE	ETBL	TRAP											
IH 2	ID 3-5	RL 4	ID 3	IH 2											

^{*} The opcode \$04 (on sheet 1 of 2) corresponds to one of the loop primitive instructions DBEQ, DBNE, IBEQ, IBNE, TBEQ, or TBNE.

Page 2: When the CPU encounters a page 2 opcode (\$18 on page 1 of the opcode map), it treats the next byte of object code as a page 2 instruction opcode.

[†] Refer to instruction summary for more information.

[‡] Refer to instruction summary for different HC12 cycle count.



Table A-3. Indexed Addressing Mode Postbyte Encoding (xb)

00	10	20	30	40	50	60	70	80	90	A0	B0	CO	D0	E0	F0
0,X	-16,X	1,+X	1,X+	0,Y	-16,Y	1,+Y	1,Y+	0,SP	-16,SP	1,+SP	1,SP+	0,PC	-16,PC	n,X	n,SP
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	9b const	9b const
01	11	21	31	41	51	61	71	81	91	A1	B1	C1	D1	E1	F1 - 0D
1,X 5b const	-15,X 5b const	2,+X pre-inc	2,X+ post-inc	1,Y 5b const	-15,Y 5b const	2,+Y pre-inc	2,Y+ post-inc	1,SP 5b const	-15,SP 5b const	2,+SP pre-inc	2,SP+ post-inc	1,PC 5b const	-15,PC 5b const	-n,X 9b const	-n,SP 9b const
02	12	22 22	32		52			82			B2	C2	D2		F2
2.X	-14.X	3.+X	3.X+	42 2.Y	-14.Y	62 3.+Y	72 3.Y+	2,SP	92 -14,SP	A2 3.+SP	3.SP+	2.PC	-14.PC	E2 n.X	n.SP
5b const	5b const	o,+∧ pre-inc	post-inc	5b const	5b const	ore-inc	post-inc	5b const	5b const	ore-inc	post-inc	5b const	5b const	16b const	16b const
03	13	23	33	43	53	63	73	83	93	A3	B3	C3	D3	E3	F3
3.X	-13.X	4.+X	4.X+	3.Y	-13.Y	4.+Y	4.Y+	3.SP	-13.SP	4.+SP	4.SP+	3.PC	-13.PC	[n,X]	[n,SP]
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	16b indr	16b indr
04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
4.X	-12.X	5.+X	5.X+	4.Y	-12.Y	5.+Y	5.Y+	4,SP	-12,SP	5.+SP	5.SP+	4.PC	-12.PC	A.X	A,SP
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	A offset	A offset
05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
5,X	-11,X	6,+X	6,X+	5,Y	-11,Y	6,+Y	6,Y+	5,SP	-11,SP	6,+SP	6,SP+	5,PC	-11,PC	B,X	B,SP
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	B offset	B offset
06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
6,X	-10,X	7,+X	7,X+	6,Y	-10,Y	7.+Y	7,Y+	6,SP	-10,SP	7,+SP	7,SP+	6,PC	-10,PC	D,X	D,SP
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	D offset	D offset
07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
7,X	-9,X	8,+X	8,X+	7,Y	-9,Y	8,+Y	8,Y+	7,SP	-9,SP	8,+SP	8,SP+	7,PC	-9,PC	[D,X]	[D,SP]
5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	pre-inc	post-inc	5b const	5b const	D indirect	D indirect
08 8.X	18 -8.X	28 8.–X	38 8.X-	48 8.Y	58 -8.Y	68 8.–Y	78 8.Y-	88	98	A8	B8	C8	D8	E8	F8
5b const	−o,∧ 5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	8,SP 5b const	–8,SP 5b const	8,-SP pre-dec	8,SP- post-dec	8,PC 5b const	-8,PC 5b const	n,Y 9b const	n,PC 9b const
09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
9.X	-7.X	7,-X	7.X-	9.Y	-7.Y	7Y	7.Y-	9.SP	-7.SP	7,-SP	7.SP-	9.PC	-7.PC	-n.Y	-n.PC
5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	9b const	9b const
OA.	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
10.X	-6.X	6X	6.X-	10.Y	-6.Y	6Y	6.Y-	10.SP	-6.SP	6SP	6,SP-	10.PC	-6.PC	n.Y	n.PC
5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	16b const	16b const
0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
11,X	-5,X	5,-X	5,X-	11,Y	-5,Y	5,-Y	5,Y-	11,SP	-5,SP	5,-SP	5,SP-	11,PC	-5,PC	[n,Y]	[n,PC]
5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	16b indr	16b indr
0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
12,X	-4,X	4,-X	4,X-	12,Y	-4,Y	4,-Y	4,Y-	12,SP	-4,SP	4,-SP	4,SP-	12,PC	-4,PC	A,Y	A,PC
5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	A offset	A offset
0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
13,X	-3,X	3,-X	3,X-	13,Y	-3,Y	3,-Y	3,Y-	13,SP	-3,SP	3,-SP	3,SP-	13,PC	-3,PC	B,Y	B,PC
5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	B offset	B offset
0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE CD	BE	CE 14 BC	DE	EE	FE
14,X 5b const	-2,X 5b const	2,-X	2,X-	14,Y 5b const	-2,Y 5b const	2,-Y	2,Y-	14,SP 5b const	-2,SP 5b const	2,-SP	2,SP-	14,PC	-2,PC 5b const	D,Y D offset	D,PC D offset
OF	1F	pre-dec 2F	post-dec 3F	4F	5F	pre-dec 6F	post-dec 7F	8F	9F	pre-dec AF	post-dec BF	5b const	DF const	EF .	FF
0F 15.X	-1.X	1X	3F 1.X-	4F 15.Y	5F -1.Y	1Y	1.Y-	15.SP	9F -1.SP	1SP	1.SP-	15.PC	-1.PC	1	ID.PC1
5b const	5b const	pre-dec	nost-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	pre-dec	post-dec	5b const	5b const	[D,Y] D indirect	D indirect
CO CONSC	OD CONST	p.e-ueu	positived	CO CONST	CO CONST	p.e-ueu	positives	CO CONST	CO CONST	pre-ueu	post-dec	OB CONST	CO CONST	5 manest	5 manect

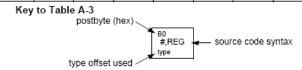


Table A-5. Transfer and Exchange Postbyte Encoding

TRANSFERS												
↓LS MS⇒	0	1	2	3	4	5	6	7				
0	A⇒A	B⇒A	CCR ⇒ A	TMP3 _L ⇒ A	B⇒A	$X_L \Rightarrow A$	Y _L ⇒A	SP _L ⇒A				
1	A⇒B	B⇒B	CCR⇒B	TMP3 _L ⇒ B	B⇒B	X _L ⇒B	Y _L ⇒B	SP _L ⇒B				
2	A⇒CCR	B⇒CCR	CCR ⇒ CCR	TMP3 _L ⇒ CCR	B ⇒ CCR	X _L ⇒CCR	Y _L ⇒CCR	SP _L ⇒ CCR				
3	sex:A ⇒ TMP2	sex:B ⇒ TMP2	sex:CCR ⇒ TMP2	TMP3 ⇒ TMP2	D ⇒ TMP2	X⇒TMP2	Y⇒TMP2	SP⇒TMP2				
4	sex:A ⇒ D SEX A,D	sex:B ⇒ D SEX B,D	sex:CCR ⇒ D SEX CCR,D	TMP3 ⇒ D	D⇒D	X⇒D	Y⇒D	SP⇒D				
5	sex:A ⇒ X SEX A,X	sex:B ⇒ X SEX B,X	sex:CCR ⇒ X SEX CCR,X	тмР3 ⇒ Х	D⇒X	X⇒X	Y⇒X	SP⇒X				
6	sex:A ⇒ Y SEX A,Y	sex:B⇒Y SEXB,Y	sex:CCR ⇒ Y SEX CCR,Y	TMP3⇒Y	D⇒Y	X⇒Y	$Y \Rightarrow Y$	SP⇒Y				
7	sex:A ⇒ SP SEX A,SP	sex:B⇒SP SEX B,SP	sex:CCR ⇒ SP SEX CCR,SP	TMP3 ⇒ SP	D⇒SP	X⇒SP	Y⇒SP	SP ⇒ SP				
			EXCH	ANGES								
↓LS MS⇒	8	9	Α	В	С	D	E	F				
0	$A \Leftrightarrow A$	B ⇔ A	CCR ⇔ A	TMP3 _L ⇒ A \$00:A ⇒ TMP3	B⇒A A⇒B	$X_L \Rightarrow A$ \$00:A $\Rightarrow X$	Y _L ⇒ A \$00:A ⇒ Y	SP _L ⇒ A \$00:A ⇒ SP				
1	A ⇔ B	B⇔B	CCR ⇔ B	$TMP3_L \Rightarrow B$ $FF:B \Rightarrow TMP3$	B⇒B \$FF⇒A	$X_L \Rightarrow B$ \$FF:B $\Rightarrow X$	$Y_L \Rightarrow B$ \$FF:B \Rightarrow Y	SP _L ⇒ B \$FF:B ⇒ SP				
2	A ⇔ CCR	B ⇔ CCR	CCR ⇔ CCR	TMP3 _L ⇒ CCR \$FF:CCR ⇒ TMP3	B ⇒ CCR \$FF:CCR ⇒ D	$X_L \Rightarrow CCR$ \$FF:CCR $\Rightarrow X$	$Y_L \Rightarrow CCR$ \$FF:CCR $\Rightarrow Y$	SP _L ⇒ CCR \$FF:CCR ⇒ SP				
3	$$00:A \Rightarrow TMP2$ $TMP2_L \Rightarrow A$	$$00:B \Rightarrow TMP2$ $TMP2_L \Rightarrow B$	\$00:CCR ⇒ TMP2 TMP2 _L ⇒ CCR	TMP3 ⇔ TMP2	D ⇔ TMP2	X ⇔ TMP2	Y ⇔ TMP2	SP ⇔ TMP2				
4	\$00:A ⇒ D	\$00:B ⇒ D	\$00:CCR⇒D B⇒CCR	TMP3 ⇔ D	D⇔D	X⇔D	Y⇔D	SP ⇔ D				
5	\$00:A ⇒ X X _L ⇒ A	\$00:B ⇒ X X _L ⇒ B	\$00:CCR ⇒ X X _L ⇒ CCR	TMP3 ⇔ X	D⇔X	X⇔X	Y⇔X	SP ⇔ X				
6	\$00:A ⇒ Y Y _L ⇒ A	\$00:B ⇒ Y Y _L ⇒ B	\$00:CCR ⇒ Y Y _L ⇒ CCR	TMP3 ⇔ Y	D⇔Y	$X \Leftrightarrow Y$	$Y \Leftrightarrow Y$	SP ⇔ Y				
7	\$00:A ⇒ SP SP _L ⇒ A	$$00:B \Rightarrow SP$ $SP_L \Rightarrow B$	\$00:CCR ⇒ SP SP _L ⇒ CCR	TMP3 ⇔ SP	D ⇔ SP	X ⇔ SP	Y ⇔ SP	SP ⇔ SP				

TMP2 and TMP3 registers are for factory use only.



Table A-6. Loop Primitive Postbyte Encoding (lb)

00 A	10 A	20 A	30 A	40 A	50 A	60 A	70 A	80 A	90 A	Ao A	Bo A
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
			(-)		(-)		/)		IBEQ.		()
(+)	(-)	(+)	1/	(+)	1.7	(+)	(-)	(+)	(-)	(+)	(-)
01 B	11 B	21 B	31 B	41 B	51 B	61 B	71 B	81 B	91 B	A1 B	B1 B
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
02	12	22	32	42	52	62	72	82	92	A2	B2
_	_	_	_	_	_	_	_	_	_	_	_
03	13	23	33	43	53	63	73	83	93	A3	B3
_	_	_	_	_	_	_	_	_	_	_	_
1											
04 D	14 D	24 D	34 D	44 D	54 D	64 D	74 D	84 D	94 D	A4 D	B4 D
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
05 X	15 X	25 X	35 X	45 X	55 X	65 X	75 X	85 X	95 X	As X	
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
06 Y	16 Y	26 Y	36 Y	46 Y	56 Y	66 Y	76 Y	86 Y	96 Y	As Y	B6 Y
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
07 SP	17 SP	27 SP	37 SP	47 SP	57 SP	67 SP	77 SP	87 SP	97 SP	A7 SP	B7 SP
DBEQ	DBEQ	DBNE	DBNE	TBEQ	TBEQ	TBNE	TBNE	IBEQ	IBEQ	IBNE	IBNE
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)

Key to Table A-6

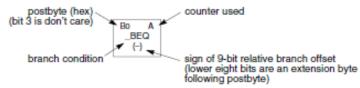


Table A-7. Branch/Complementary Branch

	Br	anch		Complementary Branch					
Test	Mnemonic	Opcode	Boolean	Test	Mnemonic	Opcode	Comment		
r>m	BGT	2E	$Z + (N \oplus V) = 0$	r≤m	BLE	2F	Signed		
r≥m	BGE	2C	$N \oplus V = 0$	r <m< td=""><td>BLT</td><td>2D</td><td>Signed</td></m<>	BLT	2D	Signed		
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Signed		
r≤m	BLE	2F	$Z + (N \oplus V) = 1$	r>m	BGT	2E	Signed		
r <m< td=""><td>BLT</td><td>2D</td><td>N ⊕ V = 1</td><td>r≥m</td><td>BGE</td><td>2C</td><td>Signed</td></m<>	BLT	2D	N ⊕ V = 1	r≥m	BGE	2C	Signed		
r>m	BHI	22	C + Z = 0	r≤m	BLS	23	Unsigned		
r≥m	BHS/BCC	24	C = 0	r <m< td=""><td>BLO/BCS</td><td>25</td><td>Unsigned</td></m<>	BLO/BCS	25	Unsigned		
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Unsigned		
r≤m	BLS	23	C + Z = 1	r>m	BHI	22	Unsigned		
r <m< td=""><td>BLO/BCS</td><td>25</td><td>C = 1</td><td>r≥m</td><td>BHS/BCC</td><td>24</td><td>Unsigned</td></m<>	BLO/BCS	25	C = 1	r≥m	BHS/BCC	24	Unsigned		
Carry	BCS	25	C = 1	No Carry	BCC	24	Simple		
Negative	BMI	2B	N = 1	Plus	BPL	2A	Simple		
Overflow	BVS	29	V = 1	No Overflow	BVC	28	Simple		
r=0	BEQ	27	Z = 1	r≠0	BNE	26	Simple		
Always	BRA	20	_	Never	BRN	21	Unconditional		

For 16-bit offset long branches precede opcode with a \$18 page prebyte.