EE4390 Microprocessors

Lesson 6,7

Instruction Set, Branch Instructions, Assembler Directives

68HC12 Instruction Set

- An instruction set is defined as a set of instructions that a microprocessor understands to execute
- 68HC12 has 209 instructions
  - group instructions into 7 basic categories

68HC12 Instruction Set (cont)

- Data transfer and manipulation
- Arithmetic - add, subtract, multiply, divide
- Logic and bit instructions
- Data test instructions
- Branch Instructions
- Function Call Instructions
- Fuzzy Logic Instructions

Data Transfer and Manipulation Instructions

- Load instruction: copies contents of specified location to specified accumulator or index register
- Affect changes to N and Z bit of CCR
- Reference Appx A

Label	**Op-Code**	**Operand(s)**	**Comment**
	LDAA	#$12	;
	LDS	#$DFFF	;

Data Transfer and Manipulation Instructions (cont)

- Store instruction: copies contents of specified accumulator or register to specified memory location

Label	**Op-Code**	**Operand(s)**	**Comment**
	STAB	$1237	;
	STD	$CFFF	;

Data Transfer and Manipulation Instructions (cont)

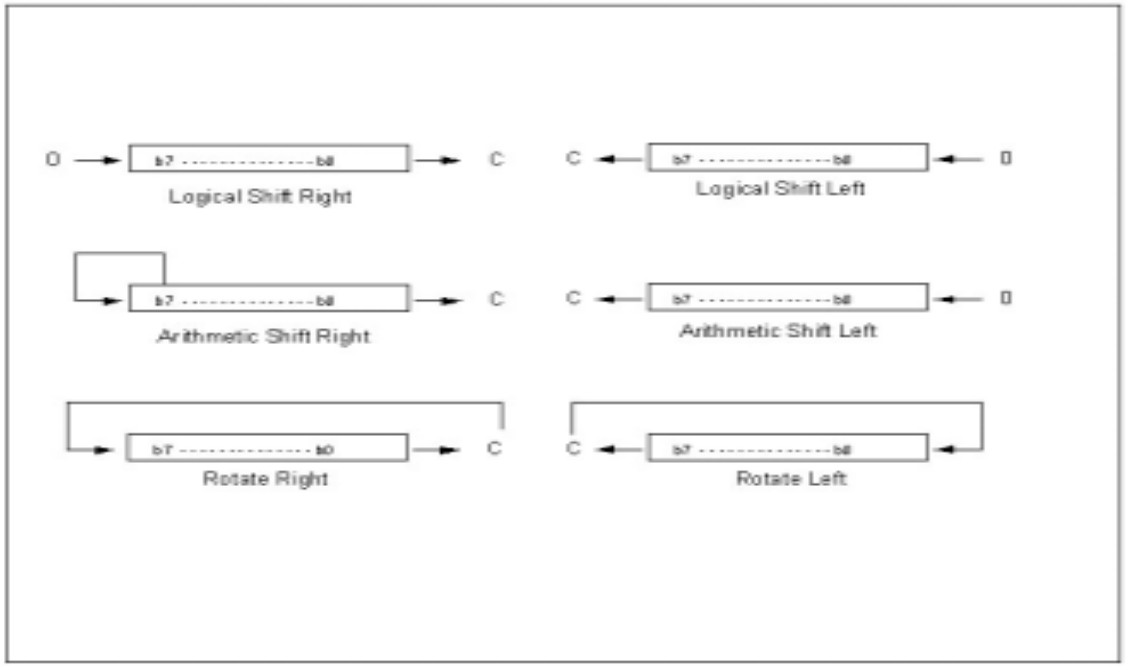
- Transfer instructions: copies the contents of one CPU register to another
- The N, Z, and V flags of the CCR are not affected

Label	**Op-Code**	**Operand(s)**	**Comment**
	TBA		;transfer B to A
	TAB		;transfer A to B
	MOVB	$1234,$CF0D	;byte mem-to-mem
		source->destination	
	MOVW	$1234,$CF0D	;word mem-to-mem
	TFR	A,B	;transfer reg to reg
		source->destination	

Rotate and Shift Operations

- 21 different rotate and shift operations
- Logical Shift
  - allows each bit to be studied by examining carry bit
  - shifts in specified direction
  - discards bit, brings in 0
- Arithmetic Shift
  - used for arithmetic multiplication and division
  - preserves sign bit
- Rotate
  - rotates bit positions within byte

Rotate and Shift Operations (cont)



Rotate and Shift Operations (cont)

Label	Op-Code	Operand(s)	Comment
	ROL	$1234	;
	RORA		
	LSLD		
	ASL	$2345	
	ASRB		

Arithmetic Operations

- Addition - 8 different instructions

Label	Op-Code	Operand(s)	Comment
	ABA		;A+B->B
	ADDB	$1234	;B + [$1234]->B

- Subtraction -

Label	Op-Code	Operand(s)	Comment
	SBA		;A-B -> A
	SUBB	$1234	;B - [$1234]->B

Arithmetic Operations (cont)

- Multiplication - 4 different instructions, integer numbers
  - MUL: multiply two 8-bit numbers in accumulator A and B and store the result in D
  - EMUL: extended multiply: multiply two 16-bit numbers in index register Y and D. The 32-bit result stored in Y and D
  - EMULS: same operation as EMUL with **signed** numbers

```
Label       Op-Code     Operand(s)    Comment
            LDAA        #$10
            LDAB        #$20
            MUL                       ;result in D
```

Arithmetic Operations (cont)

- Division - 5 different instructions, 16 and 32 bit arguments
  - IDIV: **unsigned integer** 16-bit division, dividend in D, divisor in X, quotient in X and remainder in D
  - IDIVS: **signed integer** 16-bit division, dividend in D, divisor in X, quotient in X and remainder in D
  - FDIV: **unsigned fractional** 16-bit division
  - EDIV: **unsigned integer** division, 32-bit dividend, 16-bit divisor
  - EDIVS: **signed integer** division, 32-bit dividend, 16-bit divisor

Label	**Op-Code**	**Operand(s)**	**Comment**
	LDD	#$0010	
	LDX	#$0020	
	FDIV		

Logical Operations

- AND: performs bit-by-bit AND operation

```
Label       Op-Code   Operand   Comment
            ANDA      $1234     ;A * M -> A
            BITA                ;A * M (sets CCR)
```

- OR: bit-by-bit OR operation
- EXOR: bit-by-bit EXOR operation

Complement Instructions

- NEGA: performs two’s complement of A
- NEGB: performs two’s complement of B
- COMA: performs one’s complement of A
- COMB: performs one’s complement of B

BIT Manipulation Instructions

- Four different BIT manipulation instructions:
  - BCLR: Bit Clear ;(M) * (mm) -> M
  - BSET: Bit Set ;(M) * (mm) -> M
  - Format: BSET memory location, mask

EX] BSET $D000, %01100000 ;mask specified as binary

  - BRCLR: Branch if bit(s) specified by a mask are Clear
  - BRSET: Branch if bit(s) specified by a mask are set
  - Format: BRCLR memory location, mask, branch addr

EX] BRCLR $D000, %01100000, done ;note use of label

:

done:

Data Test Instructions
Compares and Tests

- 10 test instructions, affect the CCR bits, usually followed by Branch instruction
- Compare: subtracts value from specified register and sets flags (N, Z, V, and C)
  - CMPA, CMPB, CBA, CPD, CPS, CPX, CPY
  - No effect on arguments
- Test: subtracts zero from specified register and sets flags (N,Z)
  - TST, TSTA, TSTB
  - No effects on arguments

Branch Instructions

- Can alter execution order of instructions
- Two types:
  - unconditional: BRA, JMP
  - conditional: Bxx relative-address
    - xx specifies which CCR flag is tested
    - precede branch instruction with test instruction to insure flags in CCR current
    - “higher” and “lower” for **unsigned** numbers
    - “greater” and “less” for **signed** numbers

Directives ImageCraft ICC12 assembler/compiler

.area <name> (abs) data and instructions belong to an area with the specifed name, “(abs)” allows section to contain .org statements

.org <exp> change the program counter to the address specified

_main:: indicates beginning of program

= used to equate name with numerical value

```
OPTION = $1039          ;link register name to memory addr
OPT_MASK = $80          ;use convenient name for register contents
LDAA #OPT_MASK
STAA  OPTION
```