Chapter 16

Motorola MC68HC11 Family MCU Architecture

Lesson 4

68HC11 MCU–Instruction Set

Addressing Modes

Basic Programming Features-

- 16-bit un-segmented memory with device and system registers, RAM,ROM, EEPROM all 16-bit addresses
- 256B/512B address space can be addressed by 8-bit direct address as per X’ or X at INIT(b8-b11= 0000)

Addressing Modes

Examples

Inherent → **STOP, INX (IX ← IX + 1)**

Extended → **16-bit address in the instruction – INC 1040H**

Direct → **8-bit address in BSET 20H 88H**

Immediate → **ADD #44H**

Indexed → **SBCA X (A ← A – [IX] – C)**

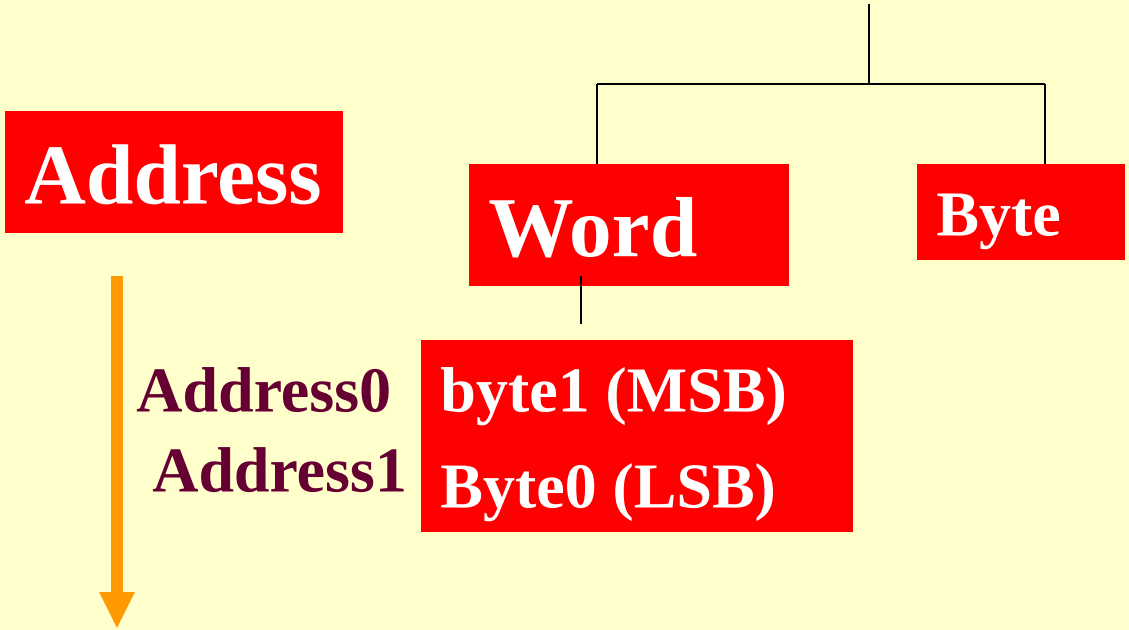
X' or X 4 bits as per init register

Data Types support

- 8-bit byte using ACCA, ACCB
- 16-bit word using IX,IY and ACCD

- A word alignment in memory is in big endian [least significant byte stored as higher bits (address 1) of a word]

Big Endian Mode



Address0- even address; Address1- odd address

16-bit Word Alignment in Memory

Data Transfer Instructions

MOV Instructions

- TAP: Transfer A to CCR ← **Affects CCR**
- TPA: Transfer CCR to A ← **Affects CCR**
- TBA: Transfer B to A ← **Affects CCR**
- TAB: Transfer from A to B

N= 0 or 1, Z= 0 or 1 V = 0

Load Instructions Using 5 Addressing Modes

- LDD: Load into ACCD
- LDX/LDY: Load into IX/IY
- LDS: Load into S

Affects CCR- N= 0 or 1, Z= 0or1
V = 0

Store Instructions Using 5 Addressing Modes

- STD: Store from ACCD
- STX/STY: Store from IX/IY
- STS: Store from S

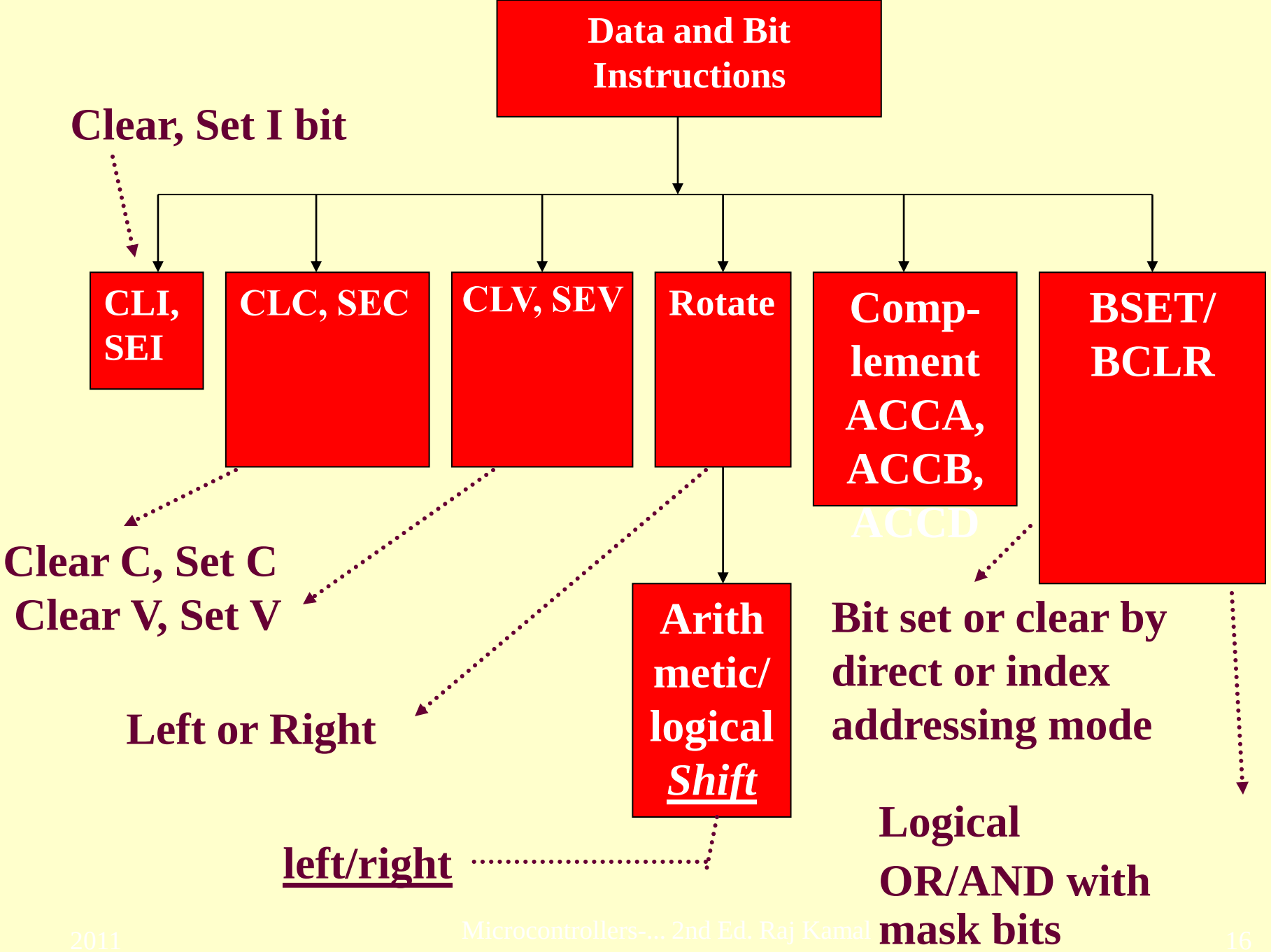
Push/POP Instructions

- PSHX/PSHY: Push IX/IY
- PULX/PULY: Pop IX/IY
- PSHA/PSHB: Push ACCA/ACCB
- PULA/PULB: Pop ACCA/ACCB

Exchange Instructions

- XGDX: ACCD ↔ IX
- XGDY: ACCD ↔ IY

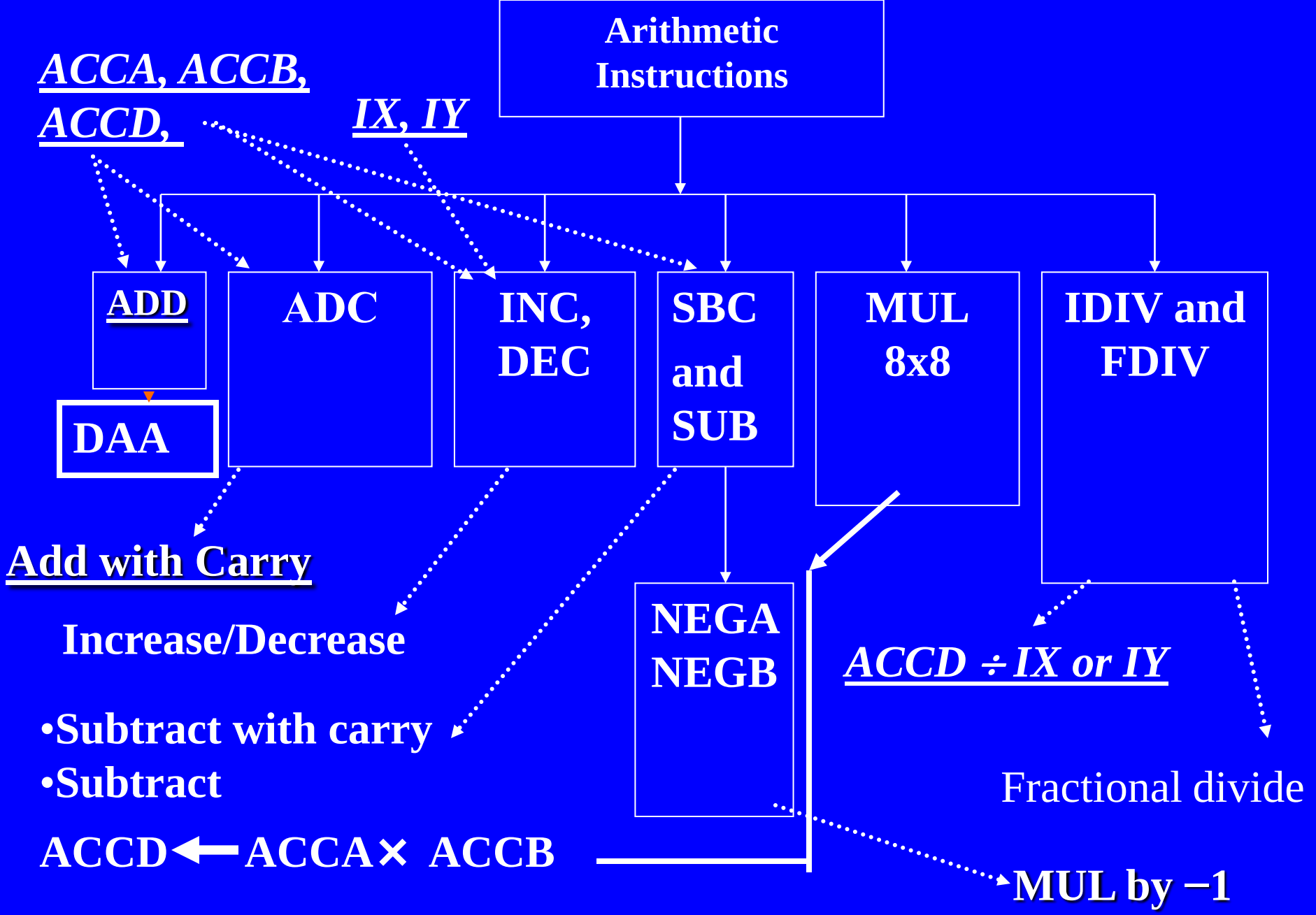
Data transfer and Bit manipulation Instructions


Data and Bit Instructions
Clear, Set I bit
CLI, SEI
CLC, SEC
CLV, SEV
Rotate
Comp-lement ACCA, ACCB, ACCD
BSET/ BCLR
Clear C, Set C
Clear V, Set V
Left or Right
Arith metic/ logical Shift
left/right
Bit set or clear by direct or index addressing mode
Logical OR/AND with mask bits

Data and Bit Manipulation Instructions

N= 0 or 1, Z= 0 or 1 and V = 0 → **Affects CCR**

Arithmetic and Logic Operation Instructions


Arithmetic Instructions
ACCA, ACCB, ACCD,
IX, IY
ADD
ADC
INC, DEC
SBC and SUB
MUL 8x8
IDIV and FDIV
DAA
Add with Carry
Increase/Decrease
•Subtract with carry
•Subtract
NEGA
NEGB
ACCD ÷ IX or IY
Fractional divide
ACCD ← ACCA × ACCB
MUL by −1

Affects on CCR

- C Affects in MUL
- N,Z,V and C Affects in Addition, Subtraction,NEG
- C,Z and V=0 Affects in DIV
- C, N and Z Affects in DAA
- ADDA,ADCA, and ADDB, ADCB ABA, ABX, ABY Affects H also

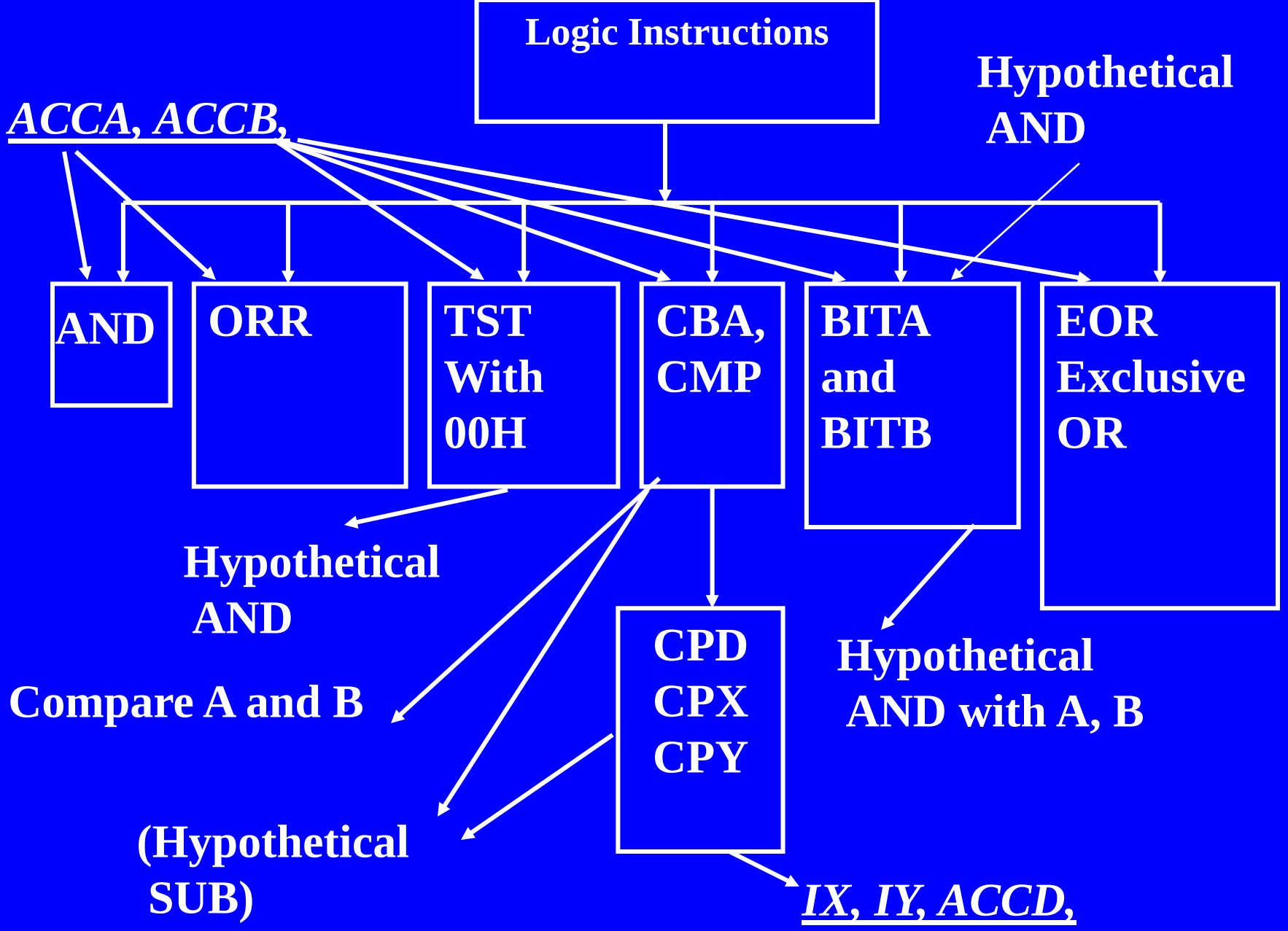
Arithmetic Instructions Using 5 Addressing Modes

- ADC: Add with Carry, ADD:Add
- SBC: Subtract with carry , SUB: Subtract
- NEG, INC, DEC
- DAA: Decimal Adjust ACC

MUL and DIV Instructions..

- MUL: Unsigned 8 x8 ACCA and ACCB ⟶ ACCD
- IDIV: Divide ***ACCD ÷ IX or IY***
- FDIV: Divide ACCD after multiplying by 10000H ***÷ IX or IY***

Fractional divide


Logic Instructions
ACCA, ACCB,
Hypothetical AND
AND
ORR
TST With 00H
CBA, CMP
BITA and BITB
EOR Exclusive OR
Hypothetical AND
Compare A and B
CPD
CPX
CPY
Hypothetical AND with A, B
(Hypothetical SUB)
IX, IY, ACCD,

Comparison and test instructions

Logic Instruction Using 5 Addressing Modes

- Logical AND, OR, EOR
- TST: Test bits (Hypothetical AND)
- CBA,CMPA, CMPB, CPD, CPX,CPY : Compare values (Hypothetical SUB)

Affects on CCR

- C,N,Z and V in Compare
- C= 0 and V = 0 N=1, Z and N in TST
- C, N and Z, V = 0 in AND, OR, EOR

Program Flow Control and Interrupt Instructions

Program Flow Control …

- Conditional Branch Instructions BRSET,BRCLR, and after comparison BLT, BLE,BEQ, BGT, BNE,BGE,BLO,BLS,BHS,BHL
- BRA: Unconditional Branch Branch to PC+ Rel
- NOP: Branch PC+1
- BRN:Branch Next Branch to PC+2

Program Flow Control …

- Conditional Branch to subroutine (BSR) PC+Label as per CCR flag
- Unconditional call to subroutine JSR
- RTS:Return from routine

SWI Software interrupt

STOP: Set CCR, stop clock, stop execution till reset or XIRQ interrupt

WAI: Wait after Pushing 9 bytes CPU registers till reset or interrupt, clock and ADC active

SWI

Pushes 9 bytes for the CPU registers to Stack

Executes if I = 1 in CCR and

The handler executes from address from an ISR vector is from 16-bits at FFF-F7H

Summary

We learnt

- big endian 16-bit data
- 8-bit byte, 16-bit word data types

We learnt

Addressing Modes

- Inherent/Register
- Extended
- Direct
- Immediate
- Index

We learnt

- Load store and data transfer instructions
- Data bit manipulation, rotate,shift

We learnt

Instructions

- ADD,ADC
- SUB,SBC, NEG
- MUL, FDIV and IDIV
- INC, DEC
- EOR,OR, AND
- Compare and Test

We learnt

Interrupt control instructions

- SWI
- STOP
- WAI

We learnt

Program flow Instructions

- BRN,
- NOP
- JSR
- BSR

End of Lesson 4 on 68HC11 Instruction Set