Microprocessor & Interfacing
Lecture 18
Arithmetic Instructions

DEPARTMENT
DRONACHARYA COLLEGE OF ENGINEERING

Contents

Introduction

- Arithmetic instruction is used for arithmetic operation such as addition subtraction multiplication and division operation. It is widely used instruction of any microprocessor and with out this instruction every microprocessor is useless.

Arithmetic Instructions

- The arithmetic instructions include
  - Addition ◦ Subtraction ◦ Multiplication ◦ Division
- Data formats
  - Unsigned binary bytes
  - Signed binary bytes
  - Unsigned binary words
  - Signed binary words
  - Unpacked decimal bytes
  - Packed decimal bytes
  - ASCII numbers

Cont..

Addition	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
Subtraction	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
Multiplication	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
Division	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

Cont..

- Addition Instructions: ADD, ADC, INC, AAA, DAA

Mnemonic	Meaning	Format	Operation	Flags Affected
ADD	Addition	ADD D, S	(S) + (D) → (D) Carry → (CF)	OF, SF, ZF, AF, PF, CF
ADC	Add with carry	ADC D, S	(S) + (D) + (CF) → (D) Carry → (CF)	OF, SF, ZF, AF, PF, CF
INC	Increment by 1	INC D	(D) + 1 → (D)	OF, SF, ZF, AF, PF
AAA	ASCII adjust for addition	AAA		AF, CF OF, SF, ZF, PF undefined
DAA	Decimal adjust for addition	DAA		SF, ZF, AF, PF, CF, OF, undefined

Cont..

Destination	Source
Register	Register
Register	Memory
Memory	Register
Register	Immediate
Memory	Immediate
Accumulator	Immediate

Allowed operands for ADD and ADC

Destination
Reg16
Reg8
Memory

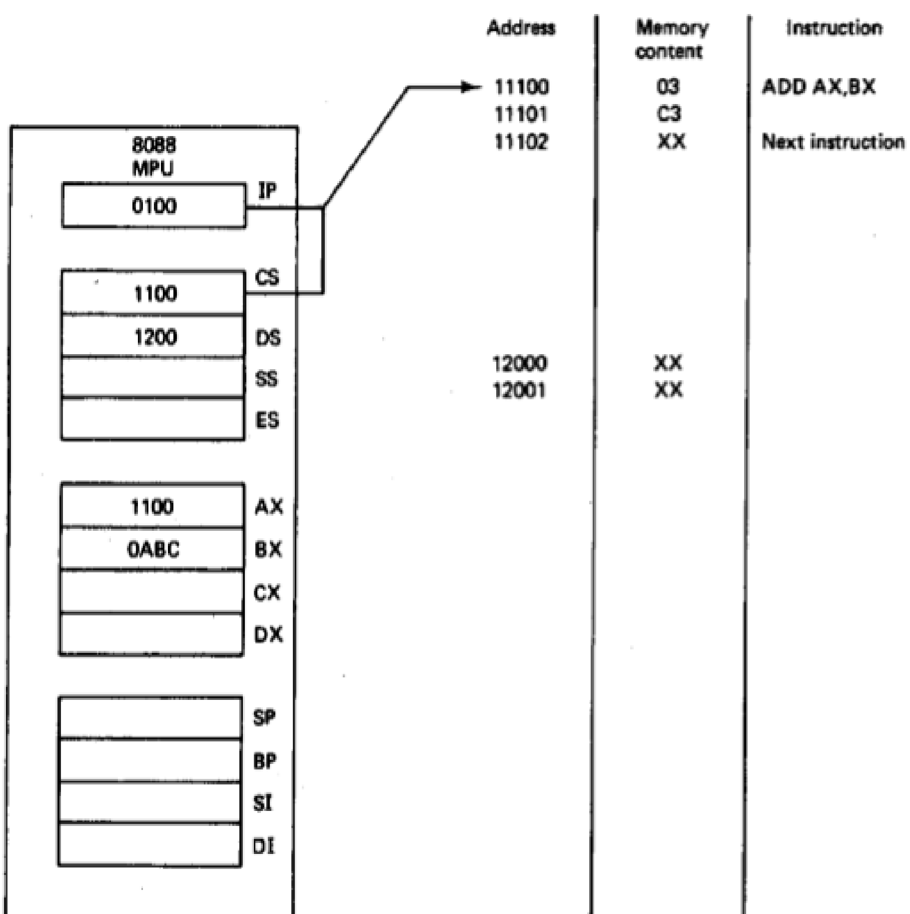
Allowed operands for INC

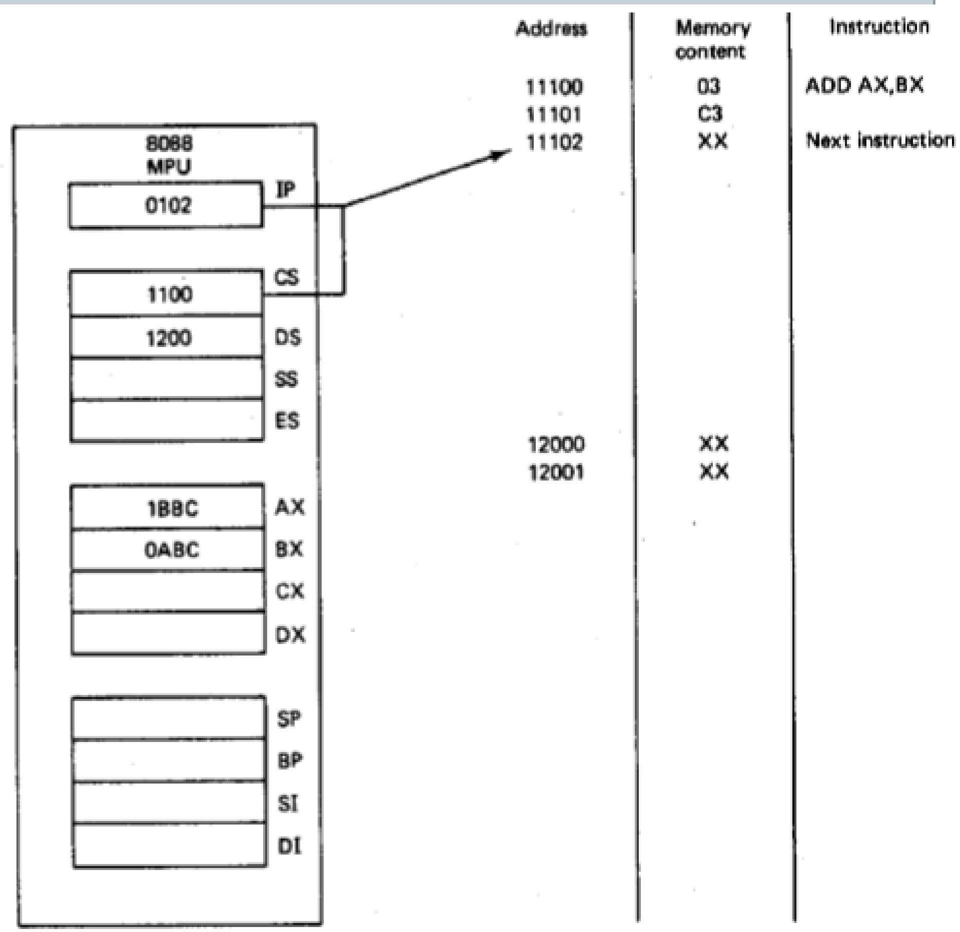
EXAMPLE

- Assume that the AX and BX registers contain 1100_{16} and $0ABC_{16}$, respectively. What is the result of executing the instruction ADD AX, BX?
- Solution:
- $(BX)+(AX)= 0ABC_{16} + 1100_{16}=1BBC_{16}$
- The sum ends up in destination register AX. That is (AX) = $1BBC_{16}$

Cont..

- Addition Instructions: ADD, ADC, INC, AAA, DAA
  - ADD AX, BX





Example

- The original contents of AX, BL, word-size memory location SUM, and carry flag (CF) are 123416, AB16, 00CD16, and 016, respectively. Describe the results of executing the following sequence of instruction?

ADD AX, [SUM]

ADC BL, 05H

INC WORD PTR [SUM]

- Solution:
- $(AX) \leftarrow (AX)+(SUM) = 1234_{16} + 00CD_{16} = 1301_{16}$
- $(BL) \leftarrow (BL)+imm8+(CF) = AB_{16} + 5_{16}+0_{16} = B0_{16}$
- $(SUM) \leftarrow (SUM)+ 1_{16} = 00CD_{16} + 1_{16} = 00CE_{16}$

Example

- What is the result of executing the following instruction sequence?

ADD AL, BL

AAA

- Assuming that AL contains 3216 (ASCII code for 2) and BL contains 3416 (ASCII code 4), and that AH has been cleared
- Solution:
- $(AL) \leftarrow (AL)+(BL)= 32_{16} + 34_{16}=66_{16}$
- The result after the AAA instruction is $(AL) = 06_{16}$ $(AH) = 00_{16}$ with both AF and CF remain cleared

Example

- Perform a 32-bit binary add operation on the contents of the processor's register.
- Solution:

(DX,CX) ← (DX,CX)+(BX,AX)

(DX,CX) = $FEDCBA98_{16}$

(BX,AX) = 01234567_{16}

```
MOV DX, 0FEDCH
MOV CX, 0BA98H
MOV BX, 01234H
MOV AX, 04567H
ADD CX, AX
ADC DX, BX ; Add with carry
```

Arithmetic Instructions

- Subtraction Instructions: SUB, SBB, DEC, AAS, DAS, and NEG

Mnemonic	Meaning	Format	Operation	Flags affected
SUB	Subtract	SUB D,S	(D) − (S) → (D) Borrow → (CF)	OF, SF, ZF, AF, PF, CF
SBB	Subtract with borrow	SBB D,S	(D) − (S) − (CF) → (D)	OF, SF, ZF, AF, PF, CF
DEC	Decrement by 1	DEC D	(D) − 1 → (D)	OF, SF, ZF, AF, PF
NEG	Negate	NEG D	0 − (D) → (D) 1 → (CF)	OF, SF, ZF, AF, PF, CF
DAS	Decimal adjust for subtraction	DAS		SF, ZF, AF, PF, CF OF undefined
AAS	ASCII adjust for subtraction	AAS		AF, CF OF, SF, ZF, PF undefined

Cont..

Destination	Source
Register	Register
Register	Memory
Memory	Register
Accumulator	Immediate
Register	Immediate
Memory	Immediate

Allowed operands for SUB and SBB instructions

Destination
Reg16
Reg8
Memory

Allowed operands for DEC instruction

Destination
Register
Memory

Allowed operands for NEG instruction

Example

- Assuming that the contents of register BX and CX are 1234_{16} and 0123_{16}, respectively, and the carry flag is 0, what is the result of executing the instruction SBB BX, CX?
- Solution:

$$(BX)-(CX)-(CF) \longrightarrow (BX)$$

We get $(BX) = 1234_{16} – 0123_{16} – 0_{16} = 1111_{16}$

- the carry flag remains cleared

Example

- Assuming that the register BX contains 003A16, what is the result of executing the following instruction?

NEG BX

Solution:

- $(BX) = 0000_{16} - (BX)$

  $= 0000_{16} + 2\text{'complement of } 003A_{16}$

  $= 0000_{16} + FFC6_{16}$

  $= FFC6_{16}$

- Since no carry is generated in this add operation, the carry flag is complemented to give (CF) = 1

Example

- Perform a 32-bit binary subtraction for variable X and Y
- Solution:

```
MOV   SI,200H        ;Initialize pointer for X
MOV   DI,100H        ;Initialize pointer for Y
MOV   AX,[SI]        ;Subtract LS words
SUB   AX,[DI]
MOV   [SI],AX        ;Save the LS word of result
MOV   AX,[SI]+2      ;Subtract MS words
SBB   AX,[DI]+2
MOV   [SI]+2,AX      ;Save the MS word of result
```

Arithmetic Instructions

- Multiplication Instructions: MUL, DIV, IMUL, IDIV, AAM, AAD, CBW, and CWD

Mnemonic	Meaning	Format	Operation	Flags Affected
MUL	Multiply (unsigned)	MUL S	(AL) · (S8) → (AX) (AX) · (S16) → (DX),(AX)	OF, CF SF, ZF, AF, PF undefined
DIV	Division (unsigned)	DIV S	(1) Q((AX)/(S8)) → (AL) R((AX)/(S8)) → (AH) (2) Q((DX,AX)/(S16)) → (AX) R((DX,AX)/(S16)) → (DX) If Q is FF_{16} in case (1) or $FFFF_{16}$ in case (2), then type 0 interrupt occurs	OF, SF, ZF, AF, PF, CF undefined
IMUL	Integer multiply (signed)	IMUL S	(AL) · (S8) → (AX) (AX) · (S16) → (DX),(AX)	OF, CF SF, ZF, AF, PF undefined
IDIV	Integer divide (signed)	IDIV S	(1) Q((AX)/(S8)) → (AL) R((AX)/(S8)) → (AH) (2) Q((DX,AX)/(S16)) → (AX) R((DX,AX)/(S16)) → (DX) If Q is positive and exceeds $7FFF_{16}$ or if Q is negative and becomes less than 8001_{16}, then type 0 interupt occurs	OF, SF, ZF, AF, PF, CF undefined

Cont..

AAM	Adjust AL for multiplication	AAM	Q((AL)/10) → (AH) R((AL)/10) → (AL)	SF, ZF, PF OF, AF,CF undefined
AAD	Adjust AX for division	AAD	(AH) · 10 + (AL) → (AL) 00 → (AH)	SF, ZF, PF OF, AF, CF undefined
CBW	Convert byte to word	CBW	(MSB of AL) → (All bits of AH)	None
CWD	Convert word to double word	CWD	(MSB of AX) → (All bits of DX)	None

Source
Reg8
Reg16
Mem8
Mem16

Example

- The 2's-complement signed data contents of AL are –1 and that of CL are –2. What result is produced in AX by executing the following instruction?

MUL CL and IMUL CL

- Solution:

(AL) = -1 (as 2's complement) = 11111111_2 = FF_{16}

(CL) = -2 (as 2's complement) = 11111110_2 = FE_{16}

Executing the MUL instruction gives

(AX) = 11111111_2 X 11111110_2 =1111110100000010_2=$FD02_{16}$

Executing the IMUL instruction gives

(AX) = -1_{16} x -2_{16} = 2_{16} = 0002_{16}

Example

- What is the result of executing the following instructions?

MOV AL, 0A1H

CBW

CWD

- Solution:

(AL) = $A1_{16}$ = 10100001_2

Executing the CBW instruction extends the MSB of AL

(AH) = 11111111_2 = FF_{16}

or (AX) = 1111111110100001_2

Executing the CWD instruction, we get

(DX) = 1111111111111111_2 = $FFFF_{16}$

That is, (AX) = $FFA1_{16}$ (DX) = $FFFF_{16}$

Scope of Research

- Design an instruction in such format that can have fast processing speed and easily understand by processor and user.