

Arithmetic and Logic Instructions

EE4380 Fall 2001

Class 7

Pari vallal Kannan

Center for Integrated Circuits and Systems

University of Texas at Dallas



Arithmetic Instructions

- Signed and Unsigned numbers
- Binary and BCD coded numbers
- Addition Instructions
 - add, addc
- Subtraction
 - subb
- Multiplication
 - mul AB
- Division
 - div AB
- BCD decimal adjust
 - da A

Unsigned Addition

- `add a, source` ; $A = A + \text{source}$
- Carry (if any) will be in CY flag
`mov A, #0F5H`
`add A, #0BH` ; $A = F5 + B0 = 00, CY=1$
- 16 bit addition
 - `addc A, source` ; $A = A + \text{source} + CY$
 - Add the lower bytes using `add`
 - Save the result
 - Add the upper bytes using `addc`

Unsigned Addition (contd.)

- Example of 16 bit addition
- Add UUVV and PPQQ

```
clr C
mov A, QQ
add A, VV
mov r6, A
mov A, PP
addc A, UU
mov r7, A
```
- Final 16 bit result in r7:r6 and CY flag

BCD Addition

- BCD – 4 bits are used to represent a decimal number from 0-9
- Packed BCD has two such numbers.
 - 17 PBCD = 17decimal = 11hex
- Packed BCD addition may not yield a valid BCD. Use decimal adjust instruction (da A) for correcting it
- After adding two Packed BCD numbers call da to get valid PBCD
 - mov A, #47H ; first BCD = 47d
 - mov B, #25H ; second BCD = 25d
 - add A,B ; A = 6CH (binary addition of 47H and 25H)
 - da A ; A = 72H (BCD result of addition)

BCD Addition (contd.)

- To correct an invalid BCD, add 6 to the digit that is greater than 9
- What da does
 - If lower nibble is > 9 or $AC=1$ then add 6 (0110) to the lower nibble
 - If upper nibble is > 9 or $CY=1$ then add 6 to the upper nibble
- da will work for ADD only. For other operations (inc, sub etc), this correction has to be done manually

Unsigned Subtraction

- subb A, source ; subtract with borrow
- To get sub, clear CY before calling subb
 - Take 2's complement of the subtrahend (source)
 - Add it to the minuend (A)
 - Invert the carry
- If the CY flag is set after the subb operation, then the result is negative and the destination has the 2's complement of the result
- subb performs subtract with borrow, if CY is set before the call. Used for 16bit subtraction

Unsigned Subtraction (contd.)

- Example

```
clr c                ; clear CY for sub operation
mov A, #4CH         ;
subb A, #6EH        ; two operands, do 4C – 6E
jnc next            ; if CY==0 result is positive
cpl A               ; CY=1, result negative. So find 2's complement
inc A               ; by complementing A and adding 1 to it
Next: mov R1, A     ; final result in R1
```

- 16 bit subtraction 2762H – 1296H

```
clr C                ; clear Cy
mov A, #62H         ;
subb A, #96H        ; 62H – 96H = CCH and CY=1
mov R7, A           ; store CC in R7
mov A, #27H         ;
subb A, #12H        ; 27H – 12H – 1 = 14H
mov R6, A           ; final result in R6:R7
```


Multiplication and Division

- **MUL AB** ; A x B, place result in BA

```
mov A, #25H
```

```
mov B, #65H
```

```
mul AB          ; 25H * 65H = E99H  
                ; B = 0EH, A = 99H
```

- **DIV AB** ; A/B, place quotient in A and remainder in B

```
mov A, #95H
```

```
mov B, #10
```

```
div AB          ; A = 9 (quotient), B = 5 (remainder)
```