

Assembly Language for Intel-Based Computers, 4th Edition

Kip R. Irvine

Unpacked and Packed Integers

Slides prepared by Kip R. Irvine

Revision date: 07/11/2002

Modified 2006, Dr. Nikolay Metodiev Sirakov

- [Chapter corrections](#) (Web) [Assembly language sources](#) (Web)

(c) Pearson Education, 2002. All rights reserved. You may modify and copy this slide show for your personal use, or for use in the classroom, as long as this copyright statement, the author's name, and the title are not changed.

ASCII and Packed Decimal Arithmetic

- Unpacked BCD
- ASCII Decimal
- AAA Instruction
- AAS Instruction
- AAM Instruction
- AAD Instruction
- Packed Decimal Integers
- DAA Instruction
- DAS Instruction

Unpacked BCD

- Binary-coded decimal (BCD) numbers use 4 binary bits to represent each decimal digit
- A number using **unpacked BCD** representation stores a decimal digit in the lower four bits of each byte
 - For example, 5,678 is stored as the following sequence of hexadecimal bytes:

05	06	07	08
----	----	----	----

ASCII Decimal

- A number using ASCII Decimal representation stores a single ASCII digit in each byte
 - For example, 5,678 is stored as the following sequence of hexadecimal bytes:

35	36	37	38
----	----	----	----

AAA Instruction

- The AAA (ASCII adjust after addition) instruction adjusts the binary result of an ADD or ADC instruction. It makes the result in AL consistent with ASCII digit representation.
 - The Carry value, if any ends up in AH
- Example: Add '8' and '2'

```
mov ah,0
mov al,'8'           ; AX = 0038h
add al,'2'           ; AX = 006Ah
aaa                  ; AX = 0100h (adjust result)
or ax,3030h         ; AX = 3130h = '10'
```

AAS Instruction

- The AAS (ASCII adjust after subtraction) instruction adjusts the binary result of an SUB or SBB instruction. It makes the result in AL consistent with ASCII digit representation.
 - It places the Carry value, if any, in AH
- Example: Subtract '9' from '8'

```
mov ah,0
mov al,'8'      ; AX = 0038h
sub al,'9'      ; AX = 00FFh
aas             ; AX = FF09h (adjust result)
pushf          ; save Carry flag
or al,30h      ; AX = FF39h (AL = '9')
popf           ; restore Carry flag
```

AAM Instruction

- The AAM (ASCII adjust after multiplication) instruction adjusts the binary result of a MUL instruction. The multiplication must have been performed on unpacked decimal numbers.

```
mov bl,05h           ; first operand
mov al,06h           ; second operand
mul bl              ; AX = 001Eh
aam                 ; AX = 0300h
```

AAD Instruction

- The AAD (ASCII adjust before division) instruction adjusts the unpacked decimal dividend in AX before a division operation

```
.data
quotient  BYTE ?
remainder BYTE ?
.code
mov ax,0307h      ; dividend
aad              ; AX = 0025h
mov bl,5         ; divisor
div bl          ; AX = 0207h
mov quotient,al
mov remainder,ah
```


Packed Decimal Integers

- **Packed BCD** stores two decimal digits per byte
 - For example, 12,345,678 can be stored as the following sequence of hexadecimal bytes:

12	34	56	78
----	----	----	----

There is no limit on the number of bytes you can use to store a BCD number. Financial values are frequently stored in BCD format, to permit high precision when performing calculations.

DAA Instruction

- The DAA (decimal adjust after addition) instruction converts the binary result of an ADD or ADC operation to packed decimal format.
- The value to be adjusted must be in AL
- Example: calculate BCD 35 + 48

```
mov al,35h
add al,48h      ; AL = 7Dh
daa            ; AL = 83h (adjusted)
```

DAS Instruction

- The DAS (decimal adjust after subtraction) instruction converts the binary result of a SUB or SBB operation to packed decimal format.
- The value must be in AL
- Example: subtract BCD 48 from 85

```
mov  al,85h
sub  al,48h      ; AL = 3Dh
das                    ; AL = 37h (adjusted)
```

The End

