

Práctica 2. Programa con instrucciones lógicas

OBJETIVO

Al terminar esta práctica, el alumno será capaz de:

Comprender el uso de las instrucciones para realizar las operaciones lógicas básicas y su efecto en el registro de banderas.

DESARROLLO

1. Use el editor de su preferencia para editar el siguiente programa en ensamblador y grábelo con el nombre de CTABITS1.ASM:

```
;*****
; CTABITS1.ASM
;
; Este programa cuenta el número de bits en uno de una
; variable de un byte. El pseudocódigo para este programa
; es:
; AL = dato
; AH = 0
; C = msb(AL) /* AL <= 1 */
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; C = msb(AL)
; AH += C
; cuenta = AH
;***** CÓDIGO DE INICIO *****
ideal
dosseg
model small
stack 256
;***** VARIABLES DEL PROGRAMA *****
dataseg
codsal db 0
dato db ?
cuenta db ?
;***** CÓDIGO DEL PROGRAMA *****
codeseg
inicio:
mov ax, @data ; Inicializa el
mov ds, ax ; segmento de datos
mov al, [dato] ; AL = dato
xor ah, ah ; AH = 0
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
```

```

adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0 ; AH += C
shl al, 1 ; C = msb(AL)
adc ah, 0
mov [cuenta], ah ; cuenta = AH
salir:
mov ah, 04Ch
mov al, [codsal]
int 21h
;***** CÓDIGO DE TERMINACIÓN *****
end inicio

```

2. Ensamble y ligue el programa anterior.

3. Cargue el programa Turbo Debugger tecleando:

td ctabits1

4. Inicialice la variable dato al valor 0AAh.

5. Haga que la Ventana de Módulo sea la ventana activa, tecleando: **[Alt] n** donde n es el número de la ventana.

6. Presione la tecla [F8] tres veces hasta ejecutar la línea: **mov al, [dato]** ; AL = dato Note que el registro AL toma el valor de dato.

7. Ejecute la siguiente línea:

xor ah, ah ; AH = 0

Observe las banderas de cero Z y de acarreo C.

Z = 1

C = 0

¿Por qué esta operación pone en cero el registro AH?

Porque al compararlo consigo mismo esta comparando los mismos bits consigo mismos, lo cual genera

8. Ejecute la siguiente línea:

shl al, 1 ; C = msb(AL),

Observe la bandera de acarreo C.

C = 1

¿A qué se debe este valor?

Se debe a que el valor mas significativo del registro al era 1, y la instruccion shl lo movio a la bandera de acarreo.

9. Ejecute la siguiente línea:

adc ah, 0 ; ah +=C

Observe el valor del registro AH:

AH = 1

¿Cómo tomó ese valor?

Tomo ese valor dado que se le sumo el valor de la bandera de acarreo (C=1), y al ser en ese momento ah=0, tomó el valor de 1

10. Repita los pasos 8 y 9 siete veces más.

11. Ejecute la siguiente línea y observe el valor de la variable cuenta. Convierte el valor de la variable dato a binario y cuente el número de bits en uno.

cuenta = 4

dato = 0AAh

¿Coincide el valor con el valor en cuenta? si coincide, el valor 0AAh, tiene 4 bits en "1".

12. Presione [Ctrl][F2] para reinicializar el programa. Vuelva a correr el programa para el valor de 5Bh.

cuenta = 5

dato =5Bh

¿Coincide el valor con el valor en cuenta? si coincide, dado que el valor 05Bh, tiene 5 bits en "1".

13. ¿Cómo modificaría este programa para que, en lugar de contar el número de unos, cuente el número de ceros? Proponga 2 opciones alternativas (Explicación + Código)

13.

OPCION 1:

CODIGO:

mov ax, @data

mov ds, ax

mov al, [dato] ;

not al ; Invierte todos los bits de AL (0s -> 1s, 1s -> 0s)

xor ah, ah ; AH = 0

shl al, 1

adc ah, 0 ; AH += c

EXPLICACION:

Se usa el not para convertir los 0's a 1's para poder contarlos mediante la carry flag,

De esta forma se pueden contar los 0's del dato, ya que estarán convertidos en 1's, todo el demás código queda igual

OPCION 2:

CODIGO:

shl al, 1

adc ah, 0 ; Esta es la última suma (Paso 8)

mov al, 8 ; Cargamos el total de bits (8) en AL

sub al, ah ; Restamos: AL = 8 - (Total de 1s en AH)

mov [cuenta], al ; Guardamos el resultado (el total de 0s)

salir:

EXPLICACION:

Al tener el programa que cuenta la cantidad de 1's podemos restarle 8 para así determinar la cantidad de ceros que contiene el dato original.