Semestre 2024-I
Estructura y Programación de Computadoras
Grupo: 2
319033993

Introducción:
Operación ADC (Add with Carry)

El ADC es una instrucción que se utiliza para realizar sumas en lenguaje ensamblador. A diferencia de una simple suma, el ADC también tiene en cuenta el valor del acarreo (carry) generado por una operación anterior. Cuando sumamos dos números y hay un acarreo del bit menos significativo (LSB), este acarreo se suma al resultado. Esto permite manejar correctamente sumas de números grandes que no caben en un solo registro.

En la tarea, se utilizará la instrucción ADC para calcular la suma de dos números hexadecimales, num1 y num2, teniendo en cuenta cualquier acarreo que pueda generarse en las operaciones.

Operación SBB (Subtract with Borrow)

El SBB, por otro lado, es una instrucción que se utiliza para realizar restas en lenguaje ensamblador. Al igual que el ADC, el SBB tiene en cuenta el valor de un préstamo (borrow) generado por una operación anterior. Cuando restamos dos números y hay un préstamo del bit menos significativo, este préstamo se resta del resultado. Esto permite manejar correctamente las restas cuando un número es mayor que el otro.

En la tarea, se utilizará la instrucción SBB para calcular la resta de num1 y num2, teniendo en cuenta cualquier préstamo que pueda generarse en las operaciones.

Almacenamiento en Memoria

Una parte crucial de la tarea es el almacenamiento de los resultados de las operaciones en memoria. Para esto, se deben utilizar las instrucciones de almacenamiento adecuadas en lenguaje ensamblador para escribir los resultados de 'suma' y 'resta' en ubicaciones de memoria específicas.

El desarrollo de este programa requerirá un profundo entendimiento de las instrucciones ADC y SBB, así como de cómo se gestionan los acarreos y préstamos en el contexto de las operaciones aritméticas. También es esencial comprender las operaciones de almacenamiento en memoria para preservar y acceder a los resultados posteriores.

El programa resultante permitirá calcular de manera precisa la suma y resta de los números hexadecimales num1 y num2, y almacenar los resultados en memoria, teniendo en cuenta las particularidades de las operaciones ADC y SBB en la arquitectura x86.

Planteamiento del Problema:
Se tienen 2 números hexadecimales:
num1 = 6790 BF93 C642 078A 1D53 40B7 E50A 75B9

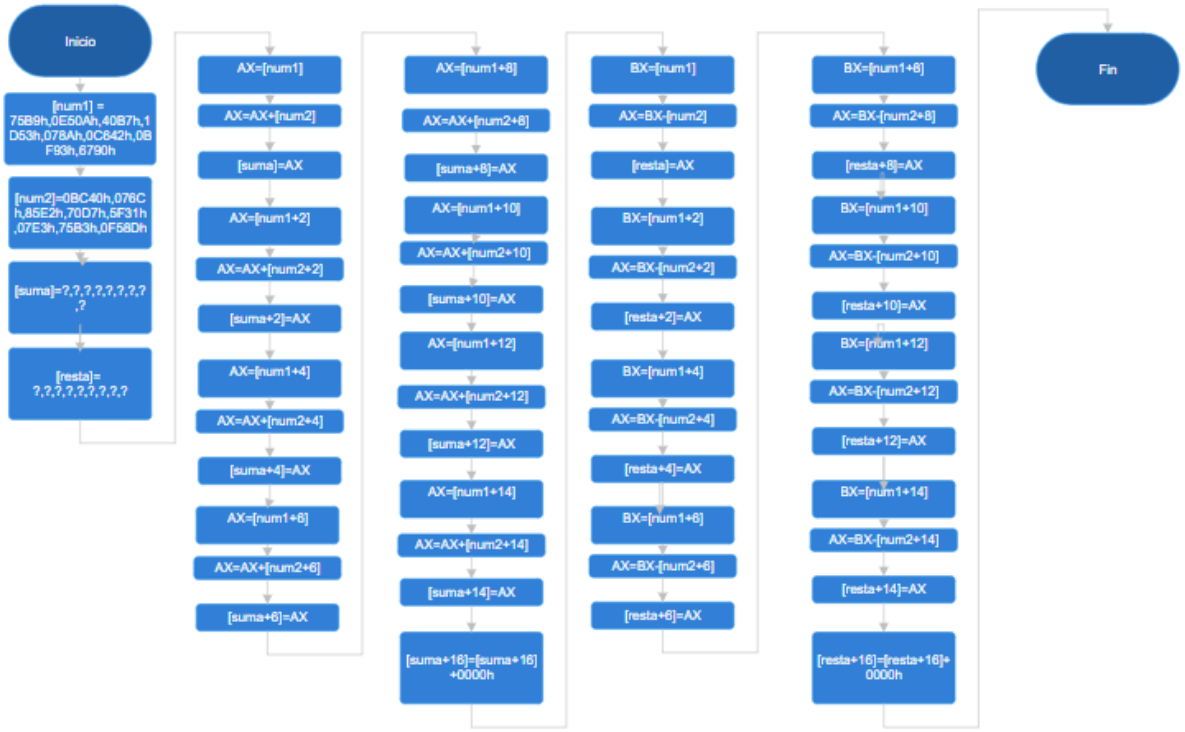
num2 = F58D 75B3 07E3 5F31 70D7 85E2 076C BC40

Desarrolle un programa en lenguaje ensamblador para arquitectura x86 que calcule :
suma = num1 + num2; y
resta = num1 - num2

Almacenar el resultado de 'suma' y 'resta' en memoria, tomando en cuenta el acarreo o el borrow según corresponda.

Para revisar la correcta ejecución del programa, debe depurarse a través de Turbo Debugger. No es necesario que se impriman los resultados en pantalla.

Diagrama de flujo



Justificación de la solución

Primero se dividen num1 y num2 en 16 bits para poder realizar la suma, dado que la arquitectura no permite operaciones más grandes. Para resolver el problema del carry y el borrow, se utilizan las instrucciones ADC y SBB. Para almacenar el resultado, se crea una localidad de memoria llamada "suma" y "resta", en las cuales se reservan 9 espacios para almacenar el resultado, junto con el carry y el borrow.

Al iniciar el programa y con la ayuda de los registros AX y BX, se copian los valores mediante la instrucción "mov" para sumarlos con ADC y luego copiar el resultado en "suma". Una vez que se ha realizado la suma de los 32 valores, se suma un 0 a [suma+16] utilizando ADC para considerar el carry. En el caso de la resta, se sigue un proceso similar, pero en lugar de sumar, se utiliza SBB. Al final, se obtienen los siguientes valores:

suma=1 5D1E 3546 CE25 66BB
resta=FFFF 7203 49E0 BE5E A858