

UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍA
DIVISION DE ELECTRONICA Y COMPUTACION

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE TRADUCTORES DE LENGUAJE I (I7026)

Reporte de Actividad Práctica
(reporte de investigación, síntesis de lectura o cuestionario)

Práctica 3: "Cálculo de potencia"

Alumno: Torres Rivera Rodrigo
Código: 397423431

Sección: D01

Profesor: Valentín Martínez López

Fecha 09 de Noviembre de 2025
firma de revisado

Nombre de la Práctica: Cálculo de potencia

Objetivo de la Práctica: Desarrollar un programa en lenguaje ensamblador que calcule la potencia de un número de 16 bits, almacenando el resultado en 32 bits (utilizando el formato Big Endian). El programa debe incluir al menos una subrutina que realice la operación de multiplicación.

Antecedentes:

Se requiere contar con el siguiente material:

- Computadora con sistema operativo compatible.
- Emulador EMU8086 instalado.

y tener conocimientos previos sobre sistemas de numeración binaria hexadecimal, así como habilidades para convertir de un sistema a otro, lo cual se complementa con conocimientos básicos de arquitectura de computadores.

Desarrollo:

Características y Requisitos:

1. Entradas:

- Crear dos variables de entrada:
 - **Base:** Un valor de 16 bits en formato **big-endian**.
 - **Exponente:** Un valor de 8 bits.

2. Salida:

- Crear una variable de salida de 32 bits en formato **Big Endian**, donde se almacenará el resultado de la operación de potencia.

3. Subrutina de multiplicación:

- Implementar una subrutina que realice la operación de multiplicación. Esta subrutina debe ser utilizada para calcular la potencia.

4. Visualización en pantalla:

- Mostrar en pantalla los nombres de las variables de entrada y salida.
- Imprimir el nombre del alumno en pantalla.

5. Finalización del programa:

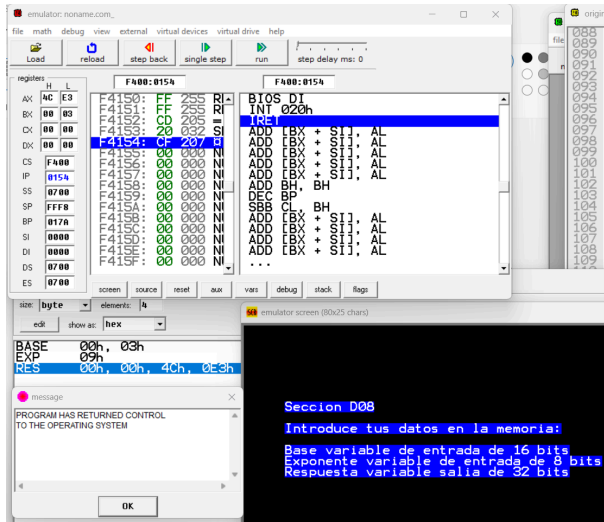
- El programa debe finalizar con la instrucción **INT 20H**.
- El registro **SP** debe tener el valor **FFF8** cuando el programa finalice correctamente. Esto asegura que los procedimientos y el retorno al sistema se realicen adecuadamente.

6. Restricciones:

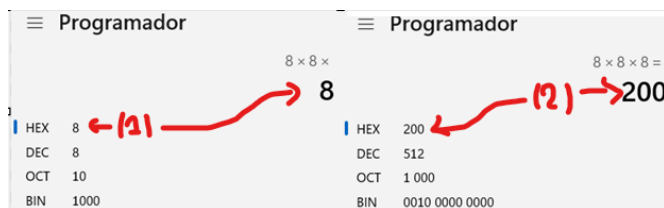
- No se permite el uso de la interrupción **21h**.

Resultados Obtenidos:

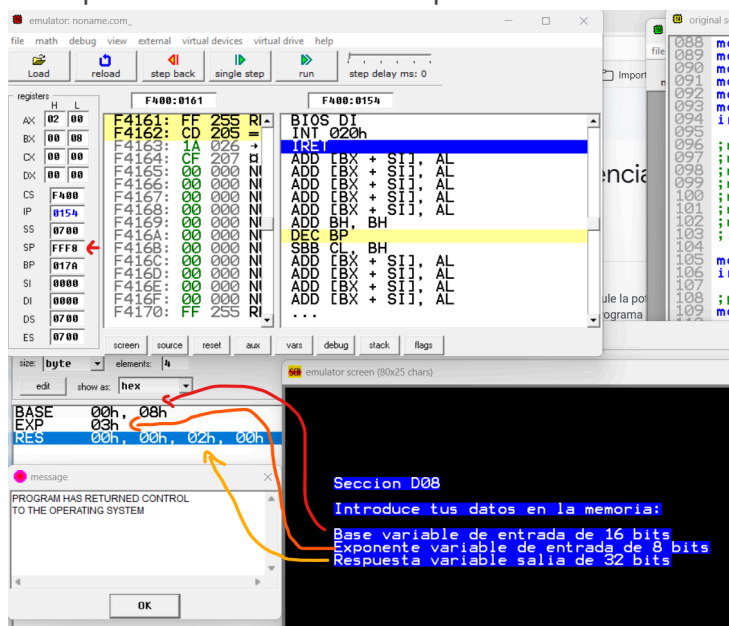
- **Entrada:**
 - Base: 0003 (16 bits)
 - Exponente: 09 (8 bits)
- **Salida:**
 - Resultado: 00004CE3 (32 bits)



Usando la calculadora en modo programador podemos verificar que el número 8 elevado al cubo (1) su resultado es 200x0H en hexadecimal (2).



Ingresando los valores en Base = 8xH Exp = 3xH el resultado es Res = 00 00 02 00 xH correspondiente a 512 en decimal para 8^3 .



Conclusiones:

Se debe tomar en cuenta el tamaño de los registros a utilizar al realizar operaciones en Lenguaje Ensamblador, así como el orden en como se guarda la información (Little Endian o Big Endian) para este caso a pesar de utilizar una variable de 32 bits para el resultado, si utilizamos base bastante grandes a un con exponentes pequeños podemos ver que el resultado se vuelve demasiado grande para que pueda quedar en la variable Res, inclusive para un registro, por esta razón la ejecución de operaciones con varios dígitos requiere de algoritmos especiales para manejarlos, como esto puede consumir mucho tiempo de cómputo se desarrolló hardware especial para estas tareas, la llama unidad de punto flotante para usarse en conjunto con los procesadores de modelo x286 y x386 llamado Co procesador matemático los modelos x287 y x387 respectivamente fueron los pares para cada modelo de procesador, actualmente está característica ya está incluida en los microprocesadores actuales.