



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e
Ingenierías

Seminario de Solución de Problemas de Traductores de Lenguajes

Nombre maestro:

Hernández Casas Ángel

Fecha:

26 / 08 / 2024

Practica:

Actividad 1

Parte I**Descripción de la Actividad**

2

Elabore una tabla comparativa de la familia de microprocesadores Intel 80x86 en orden cronológico, presentando características como la velocidad, el orden del registro, capacidad de memoria, registros y bus de datos.

Desarrollo y Resultados de la Actividad**Velocidad:**

Indica la frecuencia del reloj a la que opera el microprocesador, expresada en megahercios (MHz).

Tamaño de los Registros:

Refleja el tamaño de los registros internos del microprocesador, que son las unidades de almacenamiento de datos en el procesador. Cuanto mayor es el tamaño, más datos puede manejar en una sola operación.

Capacidad de Memoria:

Define la cantidad máxima de memoria que el procesador puede direccionar, influenciada por el tamaño del bus de direcciones.

Bus de Datos:

Representa el ancho del bus de datos, que es el canal a través del cual el microprocesador comunica datos con la memoria y otros dispositivos. Un bus más ancho puede transferir más datos simultáneamente.

Bus de Direcciones:

Refleja la cantidad de direcciones de memoria que el procesador puede manejar, lo que afecta directamente la capacidad de memoria máxima que puede utilizar.

Microprocesador	Velocidad	Tamaño de los Registros	Capacidad de Memoria	Bus de Datos	Bus de Direcciones
Intel 8086	4.77 MHz - 10 MHz	16 bits	1 MB (20 bits de direcciones)	16 bits	20 bits
Intel 80186	6 MHz - 25 MHz	16 bits	1 MB (20 bits de direcciones)	16 bits	20 bits
Intel 80286	6 MHz - 25 MHz	16 bits	16 MB (24 bits de direcciones)	16 bits	24 bits

Intel 80386	12 MHz - 40 MHz	32 bits	4 GB (32 bits de direcciones)	32 bits	32 bits
Intel 80486	20 MHz - 100 MHz	32 bits	4 GB (32 bits de direcciones)	32 bits	32 bits
Intel 80586 (Pentium)	60 MHz - 300 MHz	32 bits	4 GB (32 bits de direcciones)	64 bits (interno), 32 bits (externo)	32 bits
Intel 80686 (Pentium Pro)	150 MHz - 200 MHz	32 bits	64 GB (36 bits de direcciones)	64 bits	36 bits

Contexto Histórico:

- **Intel 8086 (1978):**

El 8086 fue el primer procesador de la serie x86 y representó un cambio importante desde los procesadores de 8 bits a los de 16 bits, permitiendo una mayor capacidad de memoria y mejores capacidades de procesamiento. Su diseño de arquitectura segmentada fue innovador para su tiempo y sentó las bases para la evolución futura de la serie x86.

- **Intel 80186 (1982):**

Basado en el 8086, el 80186 fue diseñado para mejorar el rendimiento y agregar características adicionales, como controladores de interrupciones. Aunque fue un éxito en aplicaciones embebidas, no tuvo el mismo impacto en el mercado de PCs que su predecesor.

- **Intel 80286 (1982):**

Con la introducción del modo protegido, el 80286 permitió la ejecución de sistemas operativos más complejos y mejoró la gestión de memoria. Fue el procesador principal en los sistemas IBM PC/AT y marcó el comienzo de la era de los PCs con capacidad para manejar más memoria y multitarea básica.

- **Intel 80386 (1985):**

El 386 trajo la arquitectura de 32 bits y presentó características avanzadas como el modo virtual, que permitió una mejor administración de los recursos del sistema y la capacidad de ejecutar múltiples procesos simultáneamente. Esto hizo posible el desarrollo de sistemas operativos multitarea y entornos de ejecución más complejos.

- **Intel 80486 (1989):**
El 486 introdujo la unidad de punto flotante (FPU) integrada y una arquitectura superscalar, lo que mejoró considerablemente el rendimiento del procesador. Su capacidad para ejecutar instrucciones en paralelo y su eficiencia hicieron de este procesador una opción popular para aplicaciones de alto rendimiento.
- **Intel 80586 (Pentium) (1993):**
El Pentium fue un gran avance con su diseño superscalar que permitía el procesamiento paralelo de instrucciones. Introdujo mejoras significativas en la velocidad de procesamiento y estableció la base para las futuras generaciones de procesadores de múltiples núcleos.
- **Intel 80686 (Pentium Pro) (1995):**
El Pentium Pro se centró en el mercado de servidores y estaciones de trabajo, ofreciendo un rendimiento superior en aplicaciones de alto rendimiento. Su soporte para grandes espacios de direcciones y mejor eficiencia en el procesamiento paralelo lo hicieron adecuado para tareas complejas y entornos de servidor.

Reflexión

Conclusión:

Bueno no voy a negar que yo desconocía por completo la historia de los antiguos procesadores de Intel y antes de llamarse dual o su clasificación "I3, I5, I9", me sorprendió por completo el hecho de que son los antecesores de los actuales procesado que Intel maneja en el mercado hoy en día.

Me resulto interesante la pequeña clase de historia que tuve al investigar y sacar información de los antiguos procesadores 80x86

IEEE Computer Society. (2023). *Digital Logic and Microprocessor Design*. IEEE Xplore.

Parte II

5

Investigar los tipos de datos que emplean los microprocesadores Intel 80x86. Luego responde las siguientes preguntas sobre el conjunto de instrucciones para dichos microprocesadores, por cada pregunta incluye dos instrucciones de ejemplo explicando que función realiza.

1- ¿Cuál es la función principal de las instrucciones aritméticas?

Las instrucciones aritméticas en los microprocesadores Intel 80x86 se encargan de realizar operaciones matemáticas básicas, como sumar, restar, multiplicar y dividir.

- **ADD:** Esta instrucción suma dos números.
ADD AX, BX
Suma el valor del registro BX al registro AX y almacena el resultado en AX.
- **SUB:** Resta un número de otro.
SUB AX, BX
Parecido al primero ejemplo Resta el valor del registro BX del registro AX y almacena el resultado en AX.
- **MUL:** Multiplica dos números.
- **DIV:** Divide dos números.

2 - ¿En qué consiste la conversión ASCII-BCD y en qué situación es útil?

Esta conversión es útil cuando se necesita realizar cálculos aritméticos en números que originalmente están representados como caracteres ASCII.

- **AAA (ASCII Adjust After Addition):** Ajusta el resultado de la suma de dos números BCD almacenados en ASCII para que sea un número BCD válido.
- **AAM (ASCII Adjust AX After Multiply):** Ajusta el resultado de la multiplicación de dos números BCD almacenados en ASCII.

3 - Describe la función de las instrucciones de cambio de bit.

Las instrucciones de cambio de bit se utilizan para manipular directamente los bits individuales dentro de un registro o una posición de memoria. Estas instrucciones permiten modificar, establecer, limpiar o comprobar el valor de bits específicos.

- **BT (Bit Test):** Prueba un bit específico de un registro o una posición de memoria y coloca el resultado en la bandera de acarreo (Carry Flag).
BT AX, 3
Prueba el tercer bit del registro AX y establece la bandera de acarreo si el bit es 1.
- **BTS (Bit Test and Set):** Prueba un bit específico y luego lo establece en 1.
BTS AX, 2

Prueba el segundo bit del registro AX. Si es 0, lo cambia a 1 y establece la bandera de acarreo según el valor original del bit.

4 - ¿Qué tipo de operaciones comparación se pueden realizar?

Las operaciones de comparación en los microprocesadores Intel 80x86 permiten comparar dos valores (que pueden estar en registros o en memoria) como si son iguales, mayores o menores.

- **CMP (Compare):** Compara dos operandos restando el segundo del primero, pero sin almacenar el resultado, solo afecta las banderas de estado.
- **TEST:** Realiza una operación AND entre dos operandos, afectando solo las banderas sin almacenar el resultado.

5 - Explica el propósito de las instrucciones de transferencia de datos.

Las instrucciones de transferencia de datos en los microprocesadores Intel 80x86 permiten mover datos, estas instrucciones son fundamentales para la manipulación de datos.

- **MOV:** Transfiere datos de una ubicación a otra.
MOV AX, BX
Copia el valor de BX en AX, dejando BX sin cambios.
- **PUSH:** Guarda un valor en la pila.
PUSH AX
Guarda el valor de AX en la pila para usarlo después

6 - ¿Qué función cumplen las instrucciones que operan sobre las banderas de dos estados?

Las instrucciones que operan sobre las banderas de dos estados permiten modificar o verificar ciertos estados del procesador. Las instrucciones relacionadas pueden cambiar estas banderas para influir en el flujo de control del programa.

- **CLC (Clear Carry Flag):** Limpia la bandera de acarreo, estableciéndola en 0.
Se usa cuando quieres asegurarte de que la bandera de acarreo no interfiera con las siguientes operaciones.
- **STC (Set Carry Flag):** Establece la bandera de acarreo en 1.
Se usa para marcar que hubo un acarreo en una operación o para preparar una operación que requiere un acarreo.

7 - ¿Qué tipo de operaciones de entrada/salida se pueden realizar?

Las operaciones de entrada/salida (I/O) permiten al procesador comunicarse con dispositivos externos, como teclados, impresoras o discos duros.

- **IN:** Lee un byte o una palabra desde un puerto de entrada.
IN AL, DX lee un byte desde el puerto especificado por DX y lo coloca en AL.
- **OUT:** Envía un byte o una palabra a un puerto de salida.

OUT DX, LA toma el valor en AL y lo envía al puerto especificado por DX.



8 - ¿Cuál es el propósito de las instrucciones lógicas?

Las instrucciones lógicas **AND**, **OR**, **XOR** y **NOT** permiten realizar operaciones sobre los bits de los datos.

- **AND (Conjunción lógica):** Para cada par de bits correspondientes (uno de cada número), el resultado es 1 si ambos bits son 1; de lo contrario, es 0.
- **OR (Disyunción lógica):** Para cada par de bits correspondientes, el resultado es 1 si al menos uno de los bits es 1; si ambos bits son 0, el resultado es 0.
- **XOR (Disyunción exclusiva):** El resultado es 1 si los bits correspondientes son diferentes (uno es 1 y el otro es 0); si ambos bits son iguales, el resultado es 0.
- **NOT (Negación lógica):** La operación NOT no compara dos números, sino que toma un solo número y convierte cada uno de sus bits en su opuesto: los 1 se convierten en 0, y los 0 se convierten en 1.

9 - ¿Cómo se utilizan las instrucciones de bucle?

Las instrucciones de bucle permiten repetir un bloque de código un número específico de veces o hasta que se cumpla una condición.

- **LOOP:** Decrementa el registro CX y repite el bucle si CX no es cero. LOOP salta a la etiqueta y repite el bloque de código hasta que CX llegue a 0.
- **LOOPZ/LOOPE:** Repite el bucle si CX no es cero y la bandera de cero está activada. LOOPZ hace lo mismo que LOOP, pero también verifica si la última operación resultó en cero.

10 - Explica la pila de datos y cómo se utilizan las instrucciones de pila.

La pila es una estructura de datos que funciona con el principio LIFO (Last In, First Out), es decir, el último dato en entrar es el primero en salir, son esenciales para la gestión de llamadas a funciones y la organización de datos temporales.

- **PUSH:** Coloca un valor en la cima de la pila.
- **POP:** Recupera un valor desde la cima de la pila.

11 - ¿Qué tipo de operaciones de cadena se pueden realizar?

Las operaciones de cadena permiten manipular secuencias de datos contiguos en la memoria, como copiar, comparar, buscar, y mover cadenas de bytes o palabras.

- **MOVSX/MOVSW:** Copia bytes/palabras de una cadena de origen a una cadena de destino. Copia un byte desde la dirección apuntada por "SI" a la dirección apuntada por "DI", y luego incrementa ambos punteros.

- **CMPSB/CMPSW:** Compara bytes/palabras de dos cadenas.
Compara el byte en la dirección apuntada por "SI" con el byte en la dirección apuntada por "DI".

12 - Describe la dirección entre las instrucciones de transferencia condicional e incondicional.

Las instrucciones de transferencia controlan el flujo del programa al cambiar la secuencia de ejecución. Las instrucciones de transferencia incondicional siempre realizan el salto a una nueva dirección sin condiciones previas. En cambio, las instrucciones de transferencia condicional solo realizan el salto si se cumple una condición específica, como que un valor sea cero o negativo.

- **JMP (incondicional):** Salta a una nueva dirección sin evaluar ninguna condición.
Transfiere el control del programa a la dirección especificada por etiqueta.
- **JE/JZ (condicional):** Salta a una nueva dirección si la bandera de cero está activada.
Transfiere el control si la última operación resultó en un valor cero.

13- ¿Qué función cumplen las instrucciones de conversión de tipo?

Las instrucciones de conversión de tipo permiten cambiar el formato de los datos, por ejemplo, de un entero de 8 bits a uno de 16 bits, o de una representación en punto flotante a una entera.

- **CBW (Convert Byte to Word):** Convierte un byte con signo en un word.
Toma un valor en AL (8 bits) y extiende su signo en AX (16 bits).
- **CWD (Convert Word to Double Word):** Convierte un word con signo en un double word.
Extiende el signo del valor en AX a DX:AX (32 bits).

Parte III

Conversión

Convierte los siguientes decimales enteros a binario utilizando el formato de 16 bits

1- 256

$256 \div 2 = 128$	Residuo: 0
$128 \div 2 = 64$	Residuo: 0
$64 \div 2 = 32$	Residuo: 0
$32 \div 2 = 16$	Residuo: 0
$16 \div 2 = 8$	Residuo: 0
$8 \div 2 = 4$	Residuo: 0
$4 \div 2 = 2$	Residuo: 0
$2 \div 2 = 1$	Residuo: 0
$1 \div 2 = 0$	Residuo: 1

Binario: 100000000

16 bits: 0000 0001 0000 0000

2- -128

$128 \div 2 = 64$	Residuo: 0
$64 \div 2 = 32$	Residuo: 0
$32 \div 2 = 16$	Residuo: 0
$16 \div 2 = 8$	Residuo: 0
$8 \div 2 = 4$	Residuo: 0
$4 \div 2 = 2$	Residuo: 0
$2 \div 2 = 1$	Residuo: 0
$1 \div 2 = 0$	Residuo: 1

Complemento a uno (invertir los bits): 01111111 \rightarrow 11111111 10000000

Sumar 1 para obtener el complemento a dos: 11111111 10000000

Ajustar a 16 bits: 1111 1111 1000 0000

3- 512

$512 \div 2 = 256$	Residuo: 0
$256 \div 2 = 128$	Residuo: 0
$128 \div 2 = 64$	Residuo: 0
$64 \div 2 = 32$	Residuo: 0
$32 \div 2 = 16$	Residuo: 0
$16 \div 2 = 8$	Residuo: 0
$8 \div 2 = 4$	Residuo: 0
$4 \div 2 = 2$	Residuo: 0
$2 \div 2 = 1$	Residuo: 0
$1 \div 2 = 0$	Residuo: 1

Binario: 1000000000

16 bits: 0000 0010 0000 0000

4- -256

```

256 ÷ 2 = 128   Residuo: 0
128 ÷ 2 =  64   Residuo: 0
 64 ÷ 2 =  32   Residuo: 0
 32 ÷ 2 =  16   Residuo: 0
 16 ÷ 2 =   8   Residuo: 0
  8 ÷ 2 =   4   Residuo: 0
  4 ÷ 2 =   2   Residuo: 0
  2 ÷ 2 =   1   Residuo: 0
  1 ÷ 2 =   0   Residuo: 1

```

Binario (8 bits): 100000000
 16 bits: 0000 0001 0000 0000
 complemento a uno: 1111 1110 1111 1111

```

1111 1110 1111 1111
+
-----
1111 1111 0000 0000

```

16 bits:| 1111 1111 0000 0000.

5- 1023

```

1023 ÷ 2 = 511   Residuo: 1
 511 ÷ 2 = 255   Residuo: 1
 255 ÷ 2 = 127   Residuo: 1
 127 ÷ 2 =  63   Residuo: 1
  63 ÷ 2 =  31   Residuo: 1
  31 ÷ 2 =  15   Residuo: 1
  15 ÷ 2 =   7   Residuo: 1
   7 ÷ 2 =   3   Residuo: 1
   3 ÷ 2 =   1   Residuo: 1
   1 ÷ 2 =   0   Residuo: 1

```

Binario: 111111111
 16 bits: 0000 0011 1111 1111|