



Arellano Granados Angel Mariano

218123444

Seminario de Traductores de Lenguajes I

I7026 D02

Reporte de Actividades 2

Actividades 2 – Parte 1 – 3

Actividad 2 – Parte 1

Descripción

Representar cada cifra en formato de 16 bits en el sistema numerico binario que emplea el cpu x86.

- A. 100_{10}
- B. -64_{10}
- C. 450_{10}
- D. -8_{10}
- E. 1260_{10}

Resuelva las operaciones de suma, resta y multiplicación respetando el formato de 16 bits.

- A. $100_{10} + 44_{10}$
- B. $-6_{10} + 2_{10}$
- C. $69_{10} - 70_{10}$
- D. $-10_{10} + 20_{10}$
- E. $(2_{10})(-6_{10})$
- F. $(-2_{10})(6_{10})$
- G. $(12_{10})(4_{10})$
- H. $(-5_{10})(-6_{10})$

Represente las siguientes cifras en formato de coma flotante respecto la longitud de 32 bits.

- A. 100_{10}
- B. -64.28_{10}
- C. 45.36_{10}
- D. -0.458_{10}
- E. 126_{10}

Desarrollo y Resultados

Parte 1

- A. $100_{10} = 0000\ 0000\ 0110\ 0100$
- B. $-64_{10} = 1111\ 1111\ 1100\ 0000$
- C. $450_{10} = 0000\ 0001\ 1100\ 0010$
- D. $-8_{10} = 1111\ 1111\ 1111\ 1000$
- E. $1260_{10} = 0000\ 0100\ 1110\ 1100$

Parte 2

- A. $100_{10} + 44_{10} = 0000\ 0000\ 0110\ 0100 + 0000\ 0000\ 0010\ 1100 = 0000\ 0000\ 1001\ 0000$
- B. $-6_{10} + 2_{10} = 1111\ 1111\ 1111\ 1010 + 0000\ 0000\ 0000\ 0010 = 1111\ 1111\ 1111\ 1100$

- C. $69_{10} - 70_{10} = 0000\ 0000\ 0100\ 0101 + 1111\ 1111\ 1011\ 1010 = 1111\ 1111\ 1111\ 1111$
- D. $-10_{10} + 20_{10} = 1111\ 1111\ 1111\ 0110 + 0000\ 0000\ 0001\ 0100 = 0000\ 0000\ 0000\ 1010$
- E. $(2_{10})(-6_{10}) = 1111\ 1111\ 1111\ 0100$
- F. $(-2_{10})(6_{10}) = 1111\ 1111\ 1111\ 0100$
- G. $(12_{10})(4_{10}) = 0000\ 0000\ 0011\ 0000$
- H. $(-5_{10})(-6_{10}) = 0000\ 0000\ 0001\ 1110$

Parte 3

- A. $100_{10} = 0\ 10000101\ 100100000000000000000000$
- B. $-64.28_{10} = 1\ 10000101\ 00000001000111101011100$
- C. $45.36_{10} = 0\ 10000100\ 01101010111000010100100$
- D. $-0.458_{10} = 1\ 01111101\ 11010100111111011111010$
- E. $126_{10} = 0\ 10000101\ 111110000000000000000000$

Reflexión

Conocer y dominar las operaciones y representaciones matemáticas en Binario nos ofrece una enorme ventaja a los programadores y tiene aún más peso ahora que trabajaremos en el más bajo de los lenguajes que es el ensamblador.

Actividad 2 – Parte 2

Descripción

Contestar el cuestionario.

Desarrollo y Resultados

Actividad 2 – Parte I

Registros del CPU 8086

Nombre:	Angel Mariano Arellano Granados	Código	218123444	Sección:	D-02
---------	------------------------------------	--------	-----------	----------	------

En la sintaxis “SEGMENTO: OFFSET”, el término OFFSET representa una distancia en bytes desde la dirección del segmento hasta el final del segmento. En un programa puede haber más de un segmento los cuales pueden iniciar casi en cualquier lugar de la memoria, variar en tamaño y estar en cualquier orden.

<input type="checkbox"/>	Falso.	<input checked="" type="checkbox"/>	Verdadero.	<input type="checkbox"/>	No lo sé.
--------------------------	--------	-------------------------------------	------------	--------------------------	-----------

Registro que también almacena la dirección de memoria del segmento donde se encuentran los datos. En ocasiones el CPU lo utiliza para el manejo de cadenas.

	CS		DS		SS
X	ES		Ninguno.		

¿Cuáles son los tipos de registros que incluye un microprocesador de la familia 8086?

	Registro descriptores y registros de memoria.
	Registros especiales, registros de banderas y registros apuntadores.
X	Registros de propósito general, uso específico, segmento de memoria y registros de control.
	Registros índices, registros generales y registros temporales.

Registro encargado de contener los códigos de condición junto con otras informaciones de estado como el signo, acarreo, desbordamiento, entre otras. Muestra el estado actual de la ejecución de una instrucción.

	MAR		MBR		I/O AR		I/O BR
	IP		IR	X	PSW		

Registro que especifica al dispositivo ya sea de entrada o salida.

	MAR		MBR	X	I/O AR		I/O BR
	IP		IR		PSW		

Registro que almacena la dirección de memoria del segmento donde se encuentran la pila del sistema.

	CS		DS	X	SS
	ES		Ninguno.		

Son los registros de uso específico.

	AX, BX, CX, DX.	X	SP, BP, SI, DI.		CS, DS, SS, ES.
	MAR, MBR, I/O AR, I/O BR, IP, IR, PSW.		Ninguno.		

Registro que almacena la dirección del segmento de memoria donde se encuentran los datos del programa.

	CS	X	DS		SS
	ES		Ninguno.		

A menudo este registro se utiliza para guardar la dirección base de listas de datos en la memoria.

	AX	X	BX		CX
	DX		Ninguno.		

Registro encargado de contener los datos que van a ser escritos en la memoria o los que fueron leídos en ella.

	MAR	X	MBR		I/O AR		I/O BR
	IP		IR		PSW		

Registro encargado de contener el desplazamiento con respecto al segmento de pila del tope de la del programa (Se utiliza con las instrucciones Push y Pop, y en llamadas a procedimientos: CALL y RET).

X	SP		BP		DI
	SI		Ninguno.		

Son los registros para el manejo de memoria.

<input type="checkbox"/>	AX, BX, CX, DX.	<input type="checkbox"/>	SP, BP, SI, DI.	<input checked="" type="checkbox"/>	CS, DS, SS, ES.
<input type="checkbox"/>	MAR, MBR, I/O AR, I/O BR, IP, IR, PSW.	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

A menudo este registro contiene el conteo para ciertas instrucciones de corrimientos y rotaciones, de iteraciones en el ciclo loop y operaciones repetidas de cadenas.

<input type="checkbox"/>	AX	<input type="checkbox"/>	BX	<input checked="" type="checkbox"/>	CX
<input type="checkbox"/>	DX	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Registro encargado de contener el desplazamiento con respecto al segmento extra de un elemento de una cadena o arreglo.

<input type="checkbox"/>	SP	<input type="checkbox"/>	BP	<input checked="" type="checkbox"/>	DI
<input type="checkbox"/>	SI	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

q

Registro encargado de contener el desplazamiento con respecto al segmento de datos de un elemento de un arreglo o cadena.

<input type="checkbox"/>	SP	<input type="checkbox"/>	BP	<input type="checkbox"/>	DI
<input checked="" type="checkbox"/>	SI	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Registro encargado de contener la dirección de la siguiente instrucción que se va a ejecutar.

<input type="checkbox"/>	MAR	<input type="checkbox"/>	MBR	<input type="checkbox"/>	I/O AR	<input type="checkbox"/>	I/O BR
<input checked="" type="checkbox"/>	IP	<input type="checkbox"/>	IR	<input type="checkbox"/>	PSW	<input type="checkbox"/>	

Registro encargado de contener el código de operación (OPCODE) de las instrucciones a ser ejecutadas.

<input checked="" type="checkbox"/>	IR	<input type="checkbox"/>	IP	<input type="checkbox"/>	BP
<input type="checkbox"/>	SP	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Registro que almacena la dirección del segmento de memoria donde se encuentra el código de un programa.

<input checked="" type="checkbox"/>	CS	<input type="checkbox"/>	DS	<input type="checkbox"/>	SS
<input type="checkbox"/>	ES	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Los registros del microprocesador 8086 se usan en el modo protegido en casi todas las tareas cuando se ejecutan programas. Elija la opción muestra la longitud de palabra de estos registros.

<input type="checkbox"/>	8 bits	<input checked="" type="checkbox"/>	16 bits	<input type="checkbox"/>	32 bits	<input type="checkbox"/>	64 bits
--------------------------	--------	-------------------------------------	---------	--------------------------	---------	--------------------------	---------

Son los registros de propósito general.

<input checked="" type="checkbox"/>	AX, BX, CX, DX.	<input type="checkbox"/>	SP, BP, SI, DI.	<input type="checkbox"/>	CS, DS, SS, ES.
<input type="checkbox"/>	MAR, MBR, I/O AR, I/O BR, IP, IR, PSW.	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Los segmentos (Data, Stack y Code Segment), son una parte del programa con inicio en una localidad divisible entre 10H (16 decimal), y pueden formar parte en cualquier lugar de la memoria, ocupando tanto espacio dependiendo de cuánto lo requiera el programa para su ejecución.

<input type="checkbox"/>	Falso.	<input checked="" type="checkbox"/>	Verdadero.	<input type="checkbox"/>	No lo sé.
--------------------------	--------	-------------------------------------	------------	--------------------------	-----------

Registro encargado de almacenar la dirección de memoria donde se efectuará la próxima lectura o escritura de datos.					
<input checked="" type="checkbox"/>	MAR		MBR		I/O AR
	IP		IR		PSW
Registro encargado de contener el desplazamiento con respecto al segmento de pila de datos almacenados en la pila de un programa.					
	SP	<input checked="" type="checkbox"/>	BP		DI
	SI		Ninguno.		
A menudo este registro contiene la parte más significativa de un producto después de una multiplicación; la parte más significativa del dividendo antes de la división.					
	AX		BX		CX
<input checked="" type="checkbox"/>	DX		Ninguno.		
A menudo este registro conserva el resultado temporal después de una operación aritmética o lógica.					
<input checked="" type="checkbox"/>	AX		BX		CX
	DX		Ninguno.		

Reflexión

Este formulario me ayudo a conocer el uso de varios registros de procesador 8086, como sus contrapartes o “hermanos” que son muy parecidos en funcionamiento, pero funcionando para tareas específicas.

Actividad 2 – Parte 3

Descripción

Investigar los siguientes tipos de ensambladores:

- Ensambladores cruzados
- Ensambladores residentes
- Macroensambladores
- Micro ensambladores
- Ensambladores de una fase
- Ensambladores de dos fases

Además, realice una tabla comparativa con las ventajas y desventajas de cada uno.

Desarrollo y Resultados

Ensambladores cruzados

Se denominan así los ensambladores que se utilizan en una computadora que posee un procesador diferente al que tendrán las computadoras donde va a ejecutarse el programa objeto producido.

El empleo de este tipo de traductores permite aprovechar el soporte de medios físicos (discos, impresoras, pantallas, etc.), y de programación que ofrecen las máquinas potentes para desarrollar programas que luego los van a ejecutar sistemas muy especializados en determinados tipos de tareas.

VENTAJAS	DESVENTAJAS
Aprovechar el soporte de medios físicos y de programación de máquinas potentes.	Pueden existir problemas de compatibilidad entre los procesadores.

Ensambladores residentes

Son aquellos que permanecen en la memoria principal de la computadora y cargan, para su ejecución, al programa objeto producido. Este tipo de ensamblador tiene la ventaja de que se puede comprobar inmediatamente el programa sin necesidad de transportarlo de un lugar a otro, como se hacía en cross-assembler, y sin necesidad de programas simuladores.

Sin embargo, puede presentar problemas de espacio de memoria, ya que el traductor ocupa espacio que no puede ser utilizado por el programador. Asimismo, también ocupará memoria el programa fuente y el programa objeto. Esto obliga a tener un espacio de memoria relativamente amplio.

Es el indicado para desarrollos de pequeños sistemas de control y sencillos automatismo empleando microprocesadores.

VENTAJAS	DESVENTAJAS
Permiten ejecutar inmediatamente el programa.	Deben mantenerse en la memoria principal tanto el ensamblador como el programa fuente y el programa objeto.

Macroensambladores

Son ensambladores que permiten el uso de macroinstrucciones (macros). Debido a su potencia, normalmente son programas robustos que no permanecen en memoria una vez generado el programa objeto. Puede variar la complejidad de los mismos, dependiendo de las posibilidades de definición y manipulación de las macroinstrucciones, pero normalmente son programas bastantes complejos, por lo que suelen ser ensambladores residentes.

VENTAJAS	DESVENTAJAS
Permite crear programas robustos que no cabrían en la memoria.	Son programas bastantes complejos.

Micro ensambladores

Generalmente, los procesadores utilizados en las computadoras tienen un repertorio fijo de instrucciones, es decir, que el intérprete de las mismas interpretaba de igual forma un determinado código de operación.

El programa que indica al intérprete de instrucciones de la UCP cómo debe actuar se denomina microprograma. El programa que ayuda a realizar este microprograma se llama microensamblador. Existen procesadores que permiten la modificación de sus microprogramas, para lo cual se utilizan microensambladores.

VENTAJAS	DESVENTAJAS
Son más generalizados y no están destinados a una sola arquitectura de computadora.	Tienen un repertorio fijo de instrucciones.

Ensambladores de una fase

Estos ensambladores leen una línea del programa fuente y la traducen directamente para producir una instrucción en lenguaje máquina o la ejecuta si se trata de una pseudoinstrucción. También va construyendo la tabla de símbolos a medida que van apareciendo las definiciones de variables, etiquetas, etc.

Debido a su forma de traducción, estos ensambladores obligan a definir los símbolos antes de ser empleados para que, cuando aparezca una referencia a un determinado símbolo en una instrucción, se conozca la dirección de dicho símbolo y se pueda traducir de forma correcta. Estos ensambladores son sencillos, baratos y ocupan poco espacio, pero tiene el inconveniente indicado.

VENTAJAS	DESVENTAJAS
Son sencillos, baratos y ocupan poco espacio.	Obligan a definir los símbolos antes de ser empleados.

Los ensambladores de dos fases se denominan así debido a que realizan la traducción en dos etapas. En la primera fase, leen el programa fuente y construyen una tabla de símbolos; de esta manera, en la segunda fase, vuelven a leer el programa fuente y pueden ir traduciendo totalmente, puesto que conocen la totalidad de los símbolos utilizados y las posiciones que se les ha asignado. Estos ensambladores son los más utilizados en la actualidad.

VENTAJAS	DESVENTAJAS
Muchas veces, la primera fase de un ensamblador extremo de dos fases genera el archivo de salida que luego se lee en la segunda fase.	Requiere de dos fases para escanear un archivo fuente.

Reflexión

Conocer los diferentes tipos de ensambladores nos permite tener una vista general de los usos y limitaciones de cada uno de ellos, para poder descubrir que fue lo que los convirtió en los mas o menos usados.