

Seminario de Solución de problemas de Traductores de Lenguajes I

Centro Universitario de Ciencias Exactas en
ingenierías

Universidad de Guadalajara



Maestro: Tonatiuh Hernandez Casas

Juan Antonio Pérez Juárez
Código: 215660996
Carrera: INCO

Actividad 2: Parte I

Registros del CPU 8086

Nombre:	Juan Antonio Pérez Juárez	Código	215660996	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 checked="" type="checkbox"/> Falso.	<input 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.

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

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

<input type="checkbox"/>	Registro descriptores y registros de memoria.
<input type="checkbox"/>	Registros especiales, registros de banderas y registros apuntadores.
<input checked="" type="checkbox"/>	Registros de propósito general, uso específico, segmento de memoria y registros de control.
<input type="checkbox"/>	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.

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

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 (especiales)

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

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

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.

	AX		BX		CX
	DX	X	Ninguno.		

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

	SP		BP	X	DI
	SI		Ninguno.		

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

	SP		BP		DI
X	SI		Ninguno.		

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

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

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

X	IR		IP		BP
	SP		Ninguno.		

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

X	CS		DS		SS
	ES		Ninguno.		

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

Registro encargado de contener el desplazamiento con respecto al segmento de pila de datos almacenados en la pila de un programa.

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

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.

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

A menudo este registro conserva el resultado temporal después de una operación aritmética o lógica.

X	AX		BX		CX
	DX		Ninguno.		

Actividad 2: Parte II

Tipos de ensambladores y sus características.

Realiza una tabla comparativa de los distintos tipos de ensambladores, incluya una descripción, ventajas y desventajas.

Tipo de Ensamblador	Descripción	Ventajas	Desventajas
Ensamblador Cruzado	Se denominan así a los ensambladores que se utilizan en una computadora que posee el procesador diferente al que tendrán las computadoras donde se va a ejecutar el programa objeto producido. El empleo de este tipo permite aprovechar el soporte de medios físicos 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.	Permite desarrollar código en una plataforma con más recursos y herramientas disponibles (como una computadora de escritorio) para sistemas que tienen capacidades limitadas (como microcontroladores o sistemas embebidos). Ideal para trabajar con sistemas que no pueden ejecutar un ensamblador propio debido a limitaciones de memoria, potencia de procesamiento o capacidades de almacenamiento.	Después de ensamblar el código, este debe transferirse al sistema objetivo, lo cual puede añadir complejidad y ser propenso a errores. Pueden surgir problemas relacionados con diferencias en el hardware entre el sistema de desarrollo y el sistema objetivo, especialmente si no se cuenta con simuladores o emuladores adecuados. Puede ser más complicado para desarrolladores novatos debido a la necesidad de conocer ambas plataformas (la

			del desarrollo y la del destino).
Ensamblador Residente	<p>Son aquellas que permanecen en la memoria principal de la computadora y cargar 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 crossassembler, 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.</p>	<p>Tanto el ensamblado como la ejecución ocurren en el mismo sistema, lo que elimina la necesidad de transferir el código entre plataformas.</p> <p>El proceso de desarrollo y prueba es más rápido porque no hay necesidad de mover archivos o configuraciones entre diferentes sistemas.</p>	<p>No es ideal cuando se necesita desarrollar para otros sistemas o hardware con diferentes arquitecturas, ya que solo funciona en la máquina donde reside.</p>
Macro Ensamblador	<p>Son ensambladores que permiten el uso de macroinstrucciones. Debido a su potencia, normalmente son programas robustos que no permanecen en memoria una vez</p>	<p>Las macros permiten crear bloques de código reutilizables, lo que reduce la redundancia y hace que el desarrollo sea más eficiente.</p> <p>Al utilizar macros, el código se vuelve</p>	<p>Un uso excesivo de macros puede llevar a que los programadores se vuelvan dependientes de éstas, haciendo que el código sea menos transparente y más difícil de seguir para otros</p>

	<p>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 bastante complejos.</p>	<p>más fácil de entender y mantener, ya que se pueden agrupar instrucciones repetitivas o complejas en bloques lógicos.</p> <p>Facilita el trabajo de los programadores al automatizar tareas repetitivas, lo que reduce el tiempo de escritura de código.</p>	<p>programadores o para quienes no están familiarizados con las macros utilizadas.</p>
MicroEnsamblador	<p>Al programa que indica al intérprete de instrucciones de la CPU como debe actuar se le 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.</p>	<p>Los microensambladores permiten reconfigurar o rediseñar las unidades de control de los procesadores, lo que brinda flexibilidad para cambios o mejoras sin rediseñar completamente el hardware.</p>	<p>A diferencia de los ensambladores tradicionales, el microensamblador no es adecuado para la mayoría de los desarrolladores de software, ya que se utiliza solo para diseñar o modificar el comportamiento interno del hardware.</p> <p>La creación de microcódigo y el uso de un microensamblador puede ser muy costoso en términos de tiempo y recursos, lo que lo hace impráctico para aplicaciones comerciales comunes o para el desarrollo de</p>

			software general
Ensamblador de 1 Fase	<p>Leen una línea y la traducen directamente para producir una instrucción de lenguaje máquina o la ejecuta si se trata de una pseudoinstrucción. Se construye la tabla de símbolos a medida que aparecen 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.</p>	<p>La implementación de un ensamblador de 1 fase es menos compleja, ya que no es necesario manejar la resolución de etiquetas o direcciones en una segunda pasada.</p> <p>La imposibilidad de revisar y corregir el código en una segunda pasada puede llevar a errores que son difíciles de identificar y corregir durante el ensamblaje.</p>	<p>No es adecuado para programas que requieren una resolución compleja de direcciones o dependencias, ya que no tiene la capacidad de realizar un análisis completo del código.</p>
Ensamblador de 2 Fases	<p>Realiza la traducción en dos etapas: 1° Fase leen el programa fuente y construyen la tabla de símbolos, 2° Fase vuelve a leer el programa fuente y pueden ir traduciendo totalmente pues reconocen la</p>	<p>Puede realizar ciertas optimizaciones durante la segunda fase, basándose en la información recopilada en la primera pasada.</p> <p>Al realizar una primera pasada para analizar el</p>	<p>La necesidad de realizar dos pasadas sobre el código hace que el proceso de ensamblaje sea más lento en comparación con un ensamblador de 1 fase.</p>

	totalidad de los símbolos. Estos ensambladores son más utilizados en la actualidad.	código y una segunda para ensamblarlo, puede manejar referencias adelantadas, como etiquetas o datos definidos después de ser referenciados.	
--	---	--	--

Actividad 2: Parte III

Entorno EMU8086

Nombre:	Juan Antonio Pérez Juárez	Código	215660	Sección:	D-0
		go	996	n:	2

Entre las opciones de la barra de herramientas, se localiza un opción que permite activar los dispositivos virtuales para simular los puertos físicos ¿Cuál es el nombre de esta opción?

<input type="checkbox"/>	External	<input checked="" type="checkbox"/>	Virtual devices	<input type="checkbox"/>	Virtual drive
<input type="checkbox"/>	Debug	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

La ventana principal del editor de texto cuenta con una barra de menú de Windows con la opción <assembler>

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

Los botones <aux> + <memory> que se habilitan con la opción “emulate” permiten ver el mapa de memoria.

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

Es una opción del entorno EMU86 que permite llamar al browser y explorar una gran variedad de documentos de ayuda.

<input type="checkbox"/>	New	<input type="checkbox"/>	Code examples	<input checked="" type="checkbox"/>	Quick start tutor
<input type="checkbox"/>	Recent file	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

El botón <view>, se localiza una opción denominada “options” que permite ver el contenido de los registros en distintas bases (hexadecimal, octal, ASCII, etc).

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

En el manejo de interrupciones, el EMU8086, verifica continuamente en el bit <IF> del registro “flags” si se presentó alguna, transfiriendo el control a la subrutina de atención de estas.

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

Por defecto, las interrupciones por hardware se encuentran habilitadas y cuando se produce una interrupción de este tipo, el EMU8086 lo indica con la leyenda ::”hardware interrupt”.

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

En el menú debug se puede insertar un “break point” cuando se está depurando el programa.

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

Dentro del entorno del editor se localiza un botón para habilitar la opción del emulador y el despliegue del código fuente ¿Cuál es la opción que muestra esa opción?

<input type="checkbox"/>	<compile>	<input type="checkbox"/>	<converter>	<input checked="" type="checkbox"/>	<emulate>
<input type="checkbox"/>	<calculator>	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

¿Cuál de las siguientes opciones de la barra de herramientas del menú principal del EMU8086 habilita la documentación de su entorno?

<input type="checkbox"/>	<open>	<input checked="" type="checkbox"/>	<help>	<input type="checkbox"/>	<options>
<input type="checkbox"/>	<about>	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Es una opción del entorno EMU8086 que permite escribir un nuevo código en lenguaje ensamblador con extensión ASM.

<input checked="" type="checkbox"/>	New	<input type="checkbox"/>	Code examples	<input type="checkbox"/>	Quick start tutot
<input type="checkbox"/>	Recent file	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Cuando se produce una interrupción en el CPU, se detiene la ejecución del programa que está procesando y este procede a ejecutar un subprograma asociado a la interrupción. Para ello, el CPU actualiza la dirección en sus registros CS:IP, a fin de actualizar su dirección de memoria para ejecutar el subprograma ubicado en la nueva dirección.

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

En el manejo de una interrupción, el CPU a través de los registros CS:IP se ubica en una nueva dirección para ejecutar el subprograma asociado a esta. La nueva dirección se extrae de memoria en una “Tabla de vectores de interrupción” que tiene 512 posiciones con valores CS e IP de 16 bits para cada uno.

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

En caso de elegir la opción <New> en el entorno del EMU8086 ¿Cuál es la opción a elegir para crear un programa ejecutable simple, pero con valores predefinidos para ubicar el código? Carga el código en la dirección predefinida 0000:7c00h.

<input type="checkbox"/>	BIN template	<input type="checkbox"/>	COM template	<input checked="" type="checkbox"/>	BOOT template
<input type="checkbox"/>	EXE template	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

En caso de elegir la opción <New> en el entorno del EMU8086 ¿Cuál es la opción a elegir para crear un programa ejecutable avanzado sin limitaciones de tamaño, ni de segmentos?

<input type="checkbox"/>	BIN template	<input type="checkbox"/>	COM template	<input type="checkbox"/>	BOOT template
<input checked="" type="checkbox"/>	EXE template	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

¿Cuál es la opción que permite ver el estado del registro de banderas en el procesador?

<input type="checkbox"/>	stack	<input checked="" type="checkbox"/>	flags	<input type="checkbox"/>	options
<input type="checkbox"/>	Symbol table	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

Entre las opciones de la barra de herramientas, se localiza la opción “Debug”, la cual provee herramientas para depurar los programas.

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

Las interrupciones por hardware se encuentran deshabilitadas cuando el bit IF=1.

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

La opción de <NEW> que crea archivos ejecutables con formato simple y permite un offset (desplazamiento) de 256 bytes. Sus códigos inician con la directiva ORG 100h (DOS y Windows)

<input type="checkbox"/>	BIN template	<input checked="" type="checkbox"/>	COM template	<input type="checkbox"/>	BOOT template
<input type="checkbox"/>	EXE template	<input type="checkbox"/>	Ninguno.	<input type="checkbox"/>	

LOAD, RELOAD, SINGLE STEP, STEP BACK y RUN, son botones que se encuentran bajo la barra de herramientas.

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

Reflexión

Esta actividad me gustó, bueno, realmente no. La verdad empiezo a recordar cosas del semestre pasado cuando curse la cátedra y recordé por que no me gusta el ensamblador, aunque tengo que reconocer que el programar teniendo en cuenta el tamaño de los registros así como su ubicación agrega una capa de dificultad que a pesar que no debe ser muy agradable y que creo que no podría con ella, siento que forja el carácter de un programador, no como en los lenguajes de programación de alto nivel, en los que ni te preocupas por el coste programacional que tiene tu algoritmo.

Pero creo que puedo aprender por gusto, a ver que sale de esto.

Bibliografía:

Tipos de ensambladores :: Informática. (n.d.).

<https://informatica4194.webnode.mx/contactanos/tipos-de-ensambladores/>

colaboradores de Wikipedia. (2024, April 2). Lenguaje ensamblador.

Wikipedia, La Enciclopedia Libre.

https://es.wikipedia.org/wiki/Lenguaje_ensamblador

Registros. (2016, August 9). Arquitectura De Computadoras.

<https://is603arquicom2016.wordpress.com/registros/>

