



Universidad Autónoma de
Baja California

Facultad de Ciencias
Químicas e Ingeniería

Organización de las Computadoras y Lenguaje Ensamblador

Instrucciones del Procesador 8088

pt. 3

Teoría

La arquitectura de procesador 8088 contiene varias directivas. Las directivas son comandos que son parte del syntax de procesador, no se debe confundir con el conjunto de instrucciones. La diferencia entre una instrucción y una directiva consiste en que la directiva no genera código máquina por ende no contribuye al tamaño del programa. Esta indica al ensamblador que acciones realizar durante el proceso de ensamblado. Indica cómo manejar un operando y como el ensamblador maneja el programa al igual de como los datos deben ser acomodados en la memoria. Una instrucción genera código máquina contribuyendo al tamaño del programa. Las directivas que se explicaran a continuación son DB, DW , DD, END, ENDS, ENDP, EQU, y PROC.

DB	Genera constantes de tipo byte de 8 bits, incluyendo cadenas.
DW	Genera constantes de tipo palabra de 16 bits, incluyendo cadenas.
DD	Genera constantes de tipo long word de 32 bits.
END	Finaliza el ensamblaje del último módulo del archivo.
ENDS	Indica el final de un segmento, estructura o unión.
ENDP	Indica el fin de una función.
EQU	Asigna valores a las etiquetas. Las etiquetas pueden ser utilizadas en expresiones en lugar de una constante.
PROC	PROC indica la inicialización de una función.

Desarrollo

Se procede a construir seis programas diferentes:

1. If - Consiste de dos saltos en caso de que la condición se cumpla se ejecuta cierta instrucción, en el caso contrario el programa termina con un jump que apunta a la etiqueta de salida.
2. If - Else - Se construyó con tres saltos dos para el caso if y el caso else, junto con un salto dedicado a impresión para determinar qué caso fue el que se cumplió.
3. For - El ciclo for está constituido por una variable 'index' y seis saltos. Un salto inicial para un indicador de que el ciclo ha comenzado, un salto que regresa a una comparación enfocada en la igualdad del índice con la

cantidad de ciclos deseada. Si la comparación se cumple se genera un salto a la etiqueta de terminación, caso contrario se ejecuta el ciclo un salto al la comparación. Se generó una etiqueta 'printWithSpace' lo cual imprime un espacio entre cada ciclo de la iteración y regresa al ciclo con un salto a la etiqueta 'returnToLoop'.

4. Case-Of - Utilizando la bandera 'JE' para identificar si una condición se cumple se diseña la función 'case-of', si el respectivo caso se cumple se utiliza un salto a la etiqueta reservada para el caso específico consecutivamente se utiliza un salto a la etiqueta 'print' para desplegar el resultado en pantalla y salir del programa.
5. While-Do - Se utiliza una variable 'index' para indicar la cantidad de ciclos que se completan. Etiquetas de impresión inicial y impresión con espacio como los ejemplos previos son utilizados. La condición se genera al inicio del ciclo, en caso de que se cumpla se ejecuta el salto de terminación lo contrario se decrementa el y saltó a la condición.
6. Do-while - Muy similar al ejemplo de while-do se inicia imprimiendo el inicio del bloque de código 'do', consecutivamente se genera la condición si la condición no se cumple una impresión del contenido en el sucede y se disminuye el valor almacenado, después se ejecuta un salto al inicio del ciclo, caso contrario se genera un salto a la etiqueta salida desplegando en pantalla el fin del ciclo.

Conclusión

El lenguaje ensamblador requiere un nivel de organización mayor a comparación de los lenguajes de alto nivel. El hecho de usar saltos para ir moviendo memoria para acceder a bloques de instrucciones nos da un control mayor sobre la cantidad de recursos que utilizamos y generamos programas más pequeños a comparación de un programa diseñado en un lenguaje de alto nivel. Un problema del lenguaje de bajo nivel es visible cuando la complejidad del programa aumenta la legibilidad de él se pierde, la legibilidad es una gran ventaja en los lenguajes de alto nivel ya que su abstracción es mayor requiriendo menos líneas de código para ejecutar más instrucciones la desventaja como previamente mencionado es que toman más espacio de memoria y puede que su optimización no se la mejor.