* Regular: Es decir debe ser simétrico y ortogonal es decir, deben poder combinarse.

Tipos de instrucciones

Es necesario encontrar un compromiso entre la simplicidad del hardware y del software. Un mínimo para llegar a ese compromiso se consigue con los tipos de instrucciones siguientes:

* Instrucciones de transferencia de datos
* Instrucciones aritméticas
* Instrucciones lógicas
* Instrucciones de control de flujo del programa
* Instrucciones de entrada y salida.

Los principales factores para tener en cuenta para decidir este formato son:

* Número de operaciones de la instrucción
* Modos de direccionamiento disponibles para cada operando
* Tamaño y tipos de datos soportados.

Referencia: Rodríguez-Aragón, L. (2013). Lenguaje Máquina, Instrucciones y Modos de Direccionamiento.. [online] ..Previa.uclm.es. Available at: https://previa.uclm.es/profesorado/licesio/docencia/mie/temaii.pdf [Accessed 19 Feb. 2019].

**Indirecto con registro:** La dirección del operador está almacenada en el registro especificado en la instrucción. **Sintaxis: apode (Ri)**

**Indirecto con pila:** El operador está almacenado en la cabecera de la pila. **Sintaxis: apode (SP)**

**Indirecto con memoria:** La dirección del operador está almacenada en la dirección de memoria. **Sintaxis: apode (A)**

**Indirectos con desplazamiento:** Son un conjunto de direccionamientos en los que la EA del operando se calcula sumando dos cantidades: base y desplazamiento.

**Registro-base:** La dirección efectiva del operando se calcula sumando el campo de desplazamiento al contenido del registro especificado. **Sintaxis: apode desp(Ri).**

**Relativo:** Es un direccionamiento registro-base que utiliza implícitamente el contador de programa. **Sintaxis: apode desp(PC).**

Referencia:

*Mendíaz Cuadros, J. (2010).Modos de direccionamiento y tipos de dato. Universidad Complutense de Madrid, [online] (1), pp.15-16. Available at: http://www.fdi.ucm.es/profesor/mendias/512/docs/tema4.pdf [Accessed 19 Feb. 2019].*

* *símbolos no resueltos + símbolos visibles. Y añade información para la depuración.*

**Registro Lenguaje Maquina:**

Los programas escritos en lenguajes simbólicos deben ser traducidos a lenguaje máquina antes de ser ejecutados. Este cuenta con:

* Compilador: traductor de lenguaje de alto nivel a lenguaje ensamblador,
* Enlazador (linker): Combina varios códigos objeto, resolviendo las referencias cruzadas y crea un código ejecutable.
* Cargador (loader): Lee y carga sobre memoria el código ejecutable. Inicializa registros, pilas y fija vectores de excepción.
* Expande macros y pseudo-instrucciones
* Interpreta las directivas de ensamblaje
* Fija la representación de los datos
* Traduce las instrucciones a código máquina.
* Crea el fichero objeto que a su vez: *Determina cabeceras, segmentos de código y segmentos de datos. Crea tablas de símbolos: símbolos no resueltos + símbolos visibles. Y añade información para la depuración.*

*Referencia:*

*Mendíaz Cuadros, J. (2010). Repertorio de instrucciones y formato de la instrucción maquina. Universidad Complutense de Madrid, [online] (2), pp.15-16. Available at: http://www.fdi.ucm.es/profesor/mendias/512/docs/tema5.pdf [Accessed 19 Feb. 2019].*

* de memoria donde se halla el operando: dirección electiva.

***Métodos de direccionamiento simples:***

**Inmediato:** El operador está contenido en un campo de la propia instrucción máquina. **Sintaxis: apode #A**

**Directo a registro:** El operador este contenido en un registro de la CPU. **Sintaxis: apode Ri**

**Directo a memoria (Absoluto):** La dirección efectiva del operador está especificada en la instrucción. **Sintaxis: apode A**

La forma o metologia donde se localizan los datos y modos para acceder a ellos.

Los datos que maneja una instrucción pueden estar ubicados en:

* **En la propia instrucción**: El operando está contenido en un campo de la propia instrucción máquina.
* **En el registro de la CPU:** Los registros de la CPU se pueden utilizar para almacenar temporalmente los datos.
* **En la memoria del computador:** En este caso será necesario especificar de algún modo la dirección

Conjunto de instrucciones de máquina que es capaz de ejecutar el computador. Debe ser:

* Completo: en el sentido de que se pueda construir un programa para evaluar una función computable usando una cantidad de memoria razonable y empleando un tiempo moderado.
* Eficaz: Esto significa que las funciones más necesarias deben poder realizarse usando pocas instrucciones.
* *Relativas a PC para instrucciones o Relativas a algún registro para datos.*
* Expande macros y pseudo-instrucciones
* Interpreta las directivas de ensamblaje
* Fija la representación de los datos
* Traduce las instrucciones a código máquina.
* Crea el fichero objeto que a su vez: *Determina cabeceras, segmentos de código y segmentos de datos. Crea tablas de símbolos:*

Un programa consta de una secuencia de instrucciones, cada una de las cuales especifica una acción particular. Una parte de la instrucción, llamada código de operación, señala la acción que va a ejecutarse. Muchas instrucciones contienen los datos que se usan o bien especifican donde están. Todo lo relatico a la localización de los datos sobre los que se realiza la operación (es decir, la dirección de los operandos) se denomina direccionamiento.

El formato de Instrucción cuenta con dos visiones(Lenguaje ensamblador y registro lenguaje maquina):

**Lenguaje Ensamblador:**

* Resuelve las etiquetas de instrucciones y datos, de modo que puedan ser:

**FORMATO DE INSTRUCCIÓN**

**CONJUNTO DE INSTRUCCIONES**

Es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora, con especial interés en la forma en que la unidad central de proceso (UCP) trabaja internamente y accede a las direcciones de memoria. Sin embargo, una descripción más técnica que se ha establecido durante mucho tiempo es la siguiente: “*La arquitectura de computadoras es la interfaz entre el software y el hardware*”.

**ARQUITECTURA DE COMPUTADORAS**

**TÉCNICAS DE DIRECCIONAMIENTO**