Formato de Instrucciones

Las instrucciones que puede ejecutar el CPU se conoce como conjunto de instrucciones. Cada instrucción en dicho conjunto debe contener la información necesaria para que el procesador pueda ejecutarla.

- Código de operación: especifica la operación a ser llevada a cabo (suma, transferencia, etc.). La operación se indica mediante un código binario.
- Referencia a Operandos Fuentes: la operación puede involucrar uno o más operandos fuentes. Corresponde a las "entradas" para la operación.
- Referencia al Operando Resultado: la operación puede producir un resultado, por lo tanto debe especificar donde se dejará dicho resultado
- Referencia a la siguiente instrucción: indica al CPU dónde está la siguiente instrucción a ser ejecutada. En la mayoría de los casos, la siguiente instrucción está inmediatamente después de la que se está ejecutando. En tales casos, no hay necesidad de especificar de forma explícita dónde se encuentra la siguiente instrucción.

En algunas máquinas todas las instrucciones tienen la misma longitud; en otras hay instrucciones con distintas longitudes. La longitud de las instrucciones puede ser menor, igual o mayor que el tamaño de la palabra. Exigir que todas las instrucciones posean la misma longitud hace más sencillo la decodificación pero puede desperdiciarse espacio.

En el momento que se diseña una nueva computadora se deben escoger los formatos de las instrucciones y para ello se deben tomar en cuenta varios factores.

Por un lado, si la nueva computadora llega a tener un éxito comercial, el conjunto de instrucciones podría sobrevivir 20 años o más. En este caso, la capacidad para añadir nuevas instrucciones y aprovechar otras oportunidades que surjan durante el tiempo de vida del computador tiene gran importancia.

Además, la eficiencia de un conjunto de instrucciones depende en gran medida de la tecnología con la que se va a implementar. Con el pasar del tiempo, esta tecnología puede cambiar enormemente y hacer que ciertas decisiones que se tomaron al momento del diseño no fueron las mejores.

Un criterio de diseño importante es el relativo al tamaño de las instrucciones. Un programa conformado por *n* instrucciones de 16 bits ocupa menos espacio de memoria que *n* instrucciones de 32 bits. Dada la tendencia de los precios de las memorias, este factor podría ser menos importante en el futuro. Si se minimiza mucho el tamaño de las instrucciones puede hacer que el proceso de decodificación y de sobreposición de instrucciones sea más complicado. Un motivo para disminuir el tamaño de las instrucciones es el ancho de banda de la memoria, es decir, el número de bits por segundo que la memoria puede suplir por segundo. La velocidad de los procesadores ha aumentado más rápidamente que la velocidad de las memorias, por lo tanto las memorias se transforman en cuellos de botellas en el sistema. El hecho de que las instrucciones sean más cortas implica un procesador más rápido. dado que los computadores modernos pueden ejecutar varias instrucciones en un ciclo de reloj, es imperativo traer varias instrucciones en cada ciclo de reloj, por lo tanto, el tamaño de la instrucción es importante.

Existe un compromiso entre el deseo de disponer un conjunto de instrucciones de máquina potente y la necesidad de ahorrar espacio. El programador desea más códigos de operación, más operandos, más modos de direccionamiento y mayor rango de direcciones. Al contar con más códigos de operación y más operandos se facilita la tarea del programador ya que puede escribir programas con menos instrucciones. De igual forma, más modos de direccionamiento también facilitan la tarea del programador en la implementación de ciertas funciones como por ejemplo la manipulación de tablas, etc. Además con el uso de mayor cantidad de memoria y la memoria virtual, los programadores demandan poder direccionar mayores rangos de memoria.

Otro criterio de diseño que debe tomarse en cuenta el espacio dentro de la instrucción que será destinado para expresar la operación deseada. No puede haber una máquina con 2^n operaciones e instrucciones de menos de n bits.

Otro criterio tiene que ver con el número de bits de los campos de dirección. Si se tiene una definición más fina de memoria (por bytes) se debe pagar el precio de tener direcciones más largas y por lo tanto instrucciones más largas. La definición más fina sería un direccionamiento por bits y en el otro extremo tenemos un direccionamiento por palabras muy largas.

Teniendo en cuenta estos criterios de diseño, vamos a estudiar en qué consiste un diseño de formato de instrucciones.

Un formato de instrucciones define la descripción en bits de una instrucción en término de las distintas partes que la componen. Un formato de instrucción debe

incluir un código de operación, e implícita o explícitamente, ninguno o algunos operandos. El formato debe, implícita o explícitamente, indicar el modo de direccionamiento para cada operando. Por lo general se emplean más de un formato de instrucción. Los formatos de instrucción pueden ser de tamaño fijo, es decir, todas las instrucciones poseen la misma longitud; o de formato variables.

Para una longitud de instrucción dada, existe claramente un compromiso entre el número de códigos de operación y la capacidad de direccionamiento. Mayor número de códigos de operación implican más bits en el campo del código. Esto reduce la cantidad de bits disponibles para el direccionamiento de los operandos. En cuanto a la definición de los operandos se deben considerar los siguientes factores:

- Número de modos de direccionamiento. El modo de direccionamiento puede a veces indicarse de forma explícita en la instrucción. En ciertos casos el modo puede estar relacionado con ciertos códigos de operación, en otros casos diversos modos pueden ser utilizados para el mismo código de operación. Para el último caso es necesario especificar el modo de forma explícita en la instrucción.
- Número de operandos. Menos direcciones hacen que los programas sean más largos y difíciles de comprender. Las instrucciones típicas de las máquinas actuales permiten dos operandos. Cada dirección de operando podría requerir su propio indicador de modo de direccionamiento.
- Registros frente a memoria. Una máquina debe disponer de registros para traer los datos al CPU para procesarlos. En el caso de un solo registro visible al usuario, la dirección del operando está implícita y no consume bits en la instrucción. Si se dispone de más registros visibles al usuario, sólo serán necesarios pocos bits para especificarlo en la instrucción.

- Rango de direcciones. Para referencias a memoria, el rango de direcciones que puede utilizarse está relacionado con el número de bits de direccionamiento. Esto impone una restricción severa y es por ello que raramente se utiliza modo directo.

Hay conjuntos de instrucciones que presentan la propiedad de ser ortogonales. En el caso de formato de instrucciones, este término indica que otros elementos o campos de una instrucción son independientes del código de operación. Es decir, la dirección de un operando se calcula siempre de la misma manera independientemente del código de operación. Además hay arquitecturas que trabajan con instrucciones con una única longitud (fija) y hay otras que adoptan formatos con distintas longitudes. Cuando se utilizan formatos con longitud variable, hace más fácil proporcionar un amplio repertorio de códigos de operación de longitud variable. El direccionamiento puede ser más flexible con varias combinaciones de referencias a registros y a memoria, así como de modos de direccionamiento.

Expansión de códigos de operación

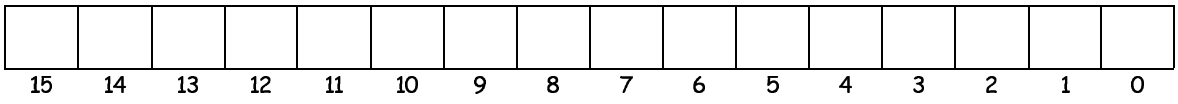
Considere una instrucción de ($n+k$) bits con un código de operación de k bits y una sola dirección de n bits. Esta instrucción permite 2^k operaciones distintas y 2^n direcciones distintas. Los mismos ($n+k$) pueden dividirse en un código de operación de ($k-1$) bits y una dirección de ($n+1$) bits, lo que implica la mitad de las operaciones y el doble de direcciones de memoria. Es posible establecer equilibrios entre los bits del código de operación y los bits de direcciones. Para ello se puede utilizar lo que se conoce como expansión o extensión de códigos.

Consideremos una máquina con instrucciones de 16 bits y las direcciones son de 4 bits. Esta situación puede ser razonable para una máquina que posee 16 registros donde se efectúan todas las operaciones. Un posible diseño sería un código de operación de 4 bits y tres direcciones en cada instrucción.

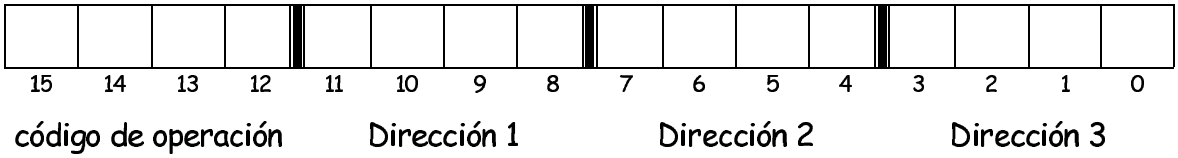
Si los diseñadores necesitan:

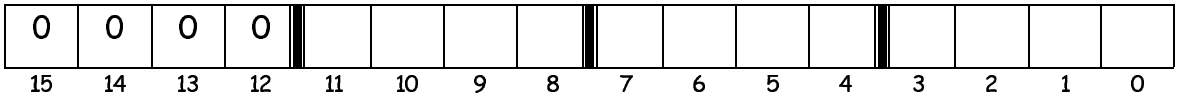
- 15 instrucciones de 3 direcciones
- 14 instrucciones de dos direcciones
- 31 instrucciones de una dirección
- 16 instrucciones de cero direcciones

que podemos hacer para permitir en 16 bits todas esas alternativas?



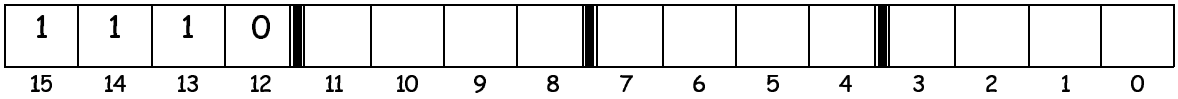
Se pueden utilizar los códigos del 0 al 14 para las instrucciones de tres direcciones y el código de operación 15 se interpreta de forma distinta.





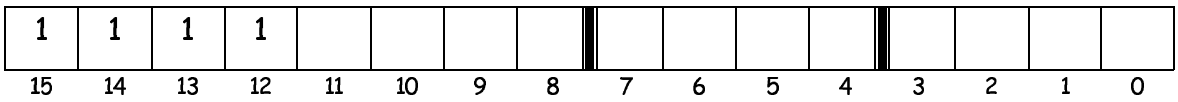
código de operación Dirección 1 Dirección 2 Dirección 3

• • •



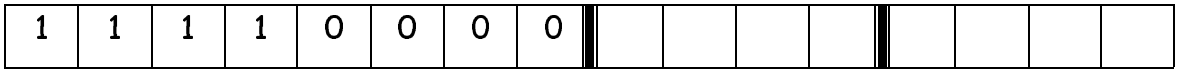
código de operación Dirección 1 Dirección 2 Dirección 3

El código 15 indica que el código de operación está contenido en los bits 8 a 15.



código de operación Dirección 1 Dirección 2

Los bits 0 a 3 corresponden a una dirección, y los bits 4 a 7 corresponden a la otra dirección. En las instrucciones de dos direcciones tendremos 1111 en los bits 12 a 15, y en el resto de los bits correspondientes al código de operación tendremos las distintas combinaciones.

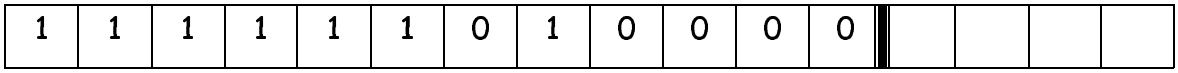


código de operación Dirección 1 Dirección 2

• • •

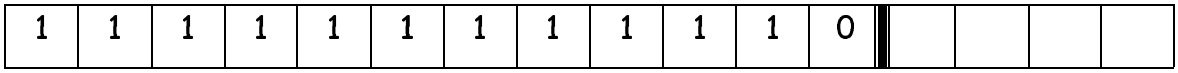
código de operación Dirección 1 Dirección 2

Las instrucciones que tienen 1111 en los bits más a la izquierda y 1110 y 1111 en los bits 8 a 11 serán tratadas de forma distinta. Estas instrucciones se tratarán como si el código de operación está en los bits 4 a 15



código de operación Dirección 1

• • •



código de operación Dirección 1

El código de operación 1111 1111 1111 se interpreta como señal de que el código de operación está en los bits 0 a 15, lo que da 16 instrucciones con cero direcciones.

Los códigos de operación fueron creciendo. Las instrucciones de tres direcciones tienen un código de operación de 4 bits; las de dos direcciones tienen un código de 8 bits, la de una dirección tienen un código de 12 bits y las de cero tienen códigos de 16 bits.

Con el uso de expansión de códigos de operación se puede minimizar la longitud promedio de las instrucciones codificando cada instrucción de modo tal que el número de bits requerido sea el mínimo. Por otro lado se puede hacer que todas las instrucciones tengan la misma longitud, asignando los códigos de operación más cortos a las instrucciones que necesiten más bits para especificar otras cosas. Se puede también minimizar el tamaño promedio de las instrucciones escogiendo códigos de operación más cortos para las instrucciones más comunes y más largos para las menos usadas.