Conjunto de Instrucciones

Todos los computadores utilizan el sistema binario internamente para su operación en lugar del sistema decimal que es el que manejamos en la vida cotidiana. En el sistema decimal utilizamos los símbolos 0,1,2,3,4,5,6,7,8,9 para representar números y cantidades, en el sistema binario sólo se utilizan los símbolos 0 y 1 que en computación reciben el nombre de bit.

Las celdas de la unidad de memoria tienen capacidad para un número fijo de dígitos binarios o bits. Esto hace que se tenga un límite en cuanto a los valores de los datos que pueden representarse en la máquina. Los registros también tienen un número fijo de bits. las instrucciones son almacenadas en la unidad de memoria y están representadas en binario al igual que los datos. Teniendo una idea muy general de cómo funciona un computador vamos a empezar a tratar en este momento acerca del conjunto de instrucciones que ofrece un computador.

Un punto de encuentro en que el diseñador del computador y el programador pueden ver la misma máquina es el conjunto de instrucciones. Desde el punto de vista del diseñador, el conjunto de instrucciones de la máquina informa de los requisitos funcionales del CPU: implementar el CPU es una tarea que, en buena parte, implica implementar el conjunto de instrucciones de máquina. Desde el punto de vista del programador, quien elige programar en un lenguaje de máquina o en lenguaje ensamblable se hace consciente de la estructura de registros y de memoria, de los tipos de datos que soporta directamente la máquina y de la funcionalidad de la ALU.

La descripción del conjunto de instrucciones de un computador es un paso más hacia la explicación del CPU.

Características de las Instrucciones de Máquina

El funcionamiento del CPU está determinado por las instrucciones que ejecuta. Estas instrucciones se denominan instrucciones de máquina. El CPU puede realizar una diversidad de funciones que son un reflejo de la variedad de las instrucciones definidas para dicho CPU. Al conjunto de instrucciones distintas que puede ejecutar el CPU se le conoce como conjunto de instrucciones o repertorio. Cuando se ejecuta una instrucción se produce un cambio de estado en el computador que puede alterar por ejemplo los valores del PC, de los registros, de la Palabra de Estado (PSW) y de la memoria.

Elementos de una instrucción de máquina

Cada instrucción debe contener la información que necesita el CPU para su ejecución.

- Código de operación. Especifica la operación a realizar (suma, resta, E/S, etc.) La operación se indica mediante un código binario.

- Referencia a operandos fuente. La operación puede involucrar a uno o más operandos fuentes, es decir, operandos que son entradas para la instrucción.

- Referencia al operando resultado. La operación puede producir un resultado.

- Referencia a la siguiente instrucción. Le indica al CPU de donde va a captar la siguiente instrucción tras completarse la ejecución de la instrucción actual. la siguiente instrucción a captar está en memoria principal. En la mayoría de los casos, la siguiente instrucción sigue inmediatamente a la instrucción en ejecución. En tales casos no hay referencia explícita a la siguiente instrucción. Cuando sea necesaria una referencia explícita, debe suministrarse la dirección de memoria.

Los operandos fuente y resultado pueden estar en alguna de las siguientes áreas:

- Memoria principal. En este caso debe indicarse la dirección de la celda que contiene el operando

- Registro del CPU. Dependiendo de la arquitectura, el CPU contiene uno o más registros que pueden ser referenciados por las instrucciones de máquina. Si sólo existe un registro, la referencia al mismo puede ser implícita. Si existe más de uno, cada registro tendrá asignado un número único, y la instrucción debe contener el número del registro deseado.

- Dispositivo de E/S. La instrucción debe especificar el módulo y dispositivo de E/S para la operación. En el caso de E/S asignadas en memoria, se dará una dirección de memoria.

Representación de las instrucciones

Cada instrucción se representa por una secuencia de bits. La instrucción está dividida en campos correspondientes a los elementos que constituyen a la misma. La descripción en campos y bits de la instrucción se conoce como formato de instrucción. En la mayoría de los conjuntos de instrucciones se emplea más de un formato. Este aspecto de representación será tratado más adelante.

Tipos de Instrucciones

El conjunto de instrucciones de máquina debe ser lo suficientemente amplio como para expresar cualquiera de las instrucciones de un lenguaje de alto nivel. Teniendo esto presente, los tipos de instrucciones se pueden clasificar de la siguiente manera:

- Procesamiento de datos

  Aritméticas

  lógicas

  Las instrucciones aritméticas proporcionan capacidad computacional para procesar datos numéricos. Las instrucciones lógicas operan sobre los bits de una palabra en lugar de considerarlos como números.

- Almacenamiento de datos/movimiento de datos. Estas instrucciones permiten la transferencia entre memoria y registros.

- E/S. estas instrucciones se necesitan para transferir programas y datos a memoria y devolver resultados de los cálculos al usuario.

- Control: comprobación y bifurcación. Estas instrucciones se emplean para comprobar el valor de una palabra de datos o el estado de un cálculo y las de bifurcación para bifurcar a diferentes conjuntos de instrucciones dependiendo de la decisión tomada.

Un aspecto importante a ser tomado en cuenta es el número de instrucciones (cardinalidad) que incluirá el conjunto de instrucciones. La tendencia es a tener un número "suficiente" para conseguir hacer bien el trabajo.

Número de direcciones en una instrucción

Una de las formas tradicionales de describir la arquitectura de un procesador es en términos del número de direcciones contenidas en cada instrucción.

Las operaciones aritméticas y lógicas son las que requieren más operandos. Prácticamente todas las operaciones aritméticas y lógicas son o bien unarios (un operando) o binarias (dos operandos). por lo tanto, necesitaríamos un máximo de dos direcciones para referenciar operandos. El resultado de una operación debe almacenarse, lo que sugiere un tercer operando. La mayoría de los CPU trabajan con instrucciones con una, dos o tres direcciones, siendo implícita la dirección de la siguiente instrucción que se mantiene en el PC.

Tres direcciones: Con instrucciones de tres direcciones se tienen dos direcciones para especificar los operandos fuentes y una dirección para el resultado

```
ADD   Y,A,B        Y=A+B
SUB   Y,A,B        Y=A-B
MULT  Y,A,B        Y=A*B
DIV   Y,A,B        Y=A/B
```

Dos direcciones: Con instrucciones de dos direcciones, y para instrucciones binarias, una de las direcciones es utilizada para especificar tanto un operando fuente como el operando resultado. Por ejemplo: X=X+Y

```
ADD   A,B          A=A+B
SUB   A,B          A=A-B
MULT  A,B          A=A*B
DIV   A,B          A=A/B
MOVE  Y,A          Y=A
```

Una dirección: las operaciones binarias tienen dos direcciones implícitas y utilizan un registro especial conocido como acumulador. Por ejemplo: Acc=Acc + X

```
ADD    A           ACC=ACC+A
SUB    A           ACC=ACC-A
MULT   A           ACC=ACC*A
DIV    A           ACC=ACC/A
LOAD   A           ACC=A
STORE  A           A=ACC
```

Cero direcciones: todas las direcciones son implícitas. Las operaciones están basadas en pilas (push, pop). El ejemplo anterior hay que convertirlo en notación postfija

```
PUSH A          PILA[TOPE]=A    TOPE=TOPE+1
POP  A          TOPE=TOPE-1     A=PILA[TOPE]
OP              PILA[TOPE-2]=PILA[TOPE-1] OP PILA[TOPE-2]
                TOPE=TOPE-1
```

Ejemplo: Y=(A-B)/(C+D*E)

3-direcciones	2-direcciones	1-dirección	0-direcciones
SUB Y,A,B	MOVE Y,A	LOAD D	PUSH E
MULT T,D,E	SUB Y,B	MULT E	PUSH D
ADD T,C,T	MOVE T,D	STORE T	MULT
DIV Y,Y,T	MULT T,E	LOAD C	PUSH C
	MOVE T1,C	ADD T	ADD
	ADD T1,T	STORE T	PUSH B
	DIV Y,T1	LOAD A	PUSH A
		SUB B	SUB
		DIV T	DIV
			POP Y

El número de direcciones por instrucción es una decisión básica de diseño. Menos direcciones por instrucción significa instrucciones más primitivas, lo que requiere un CPU menos complejo. También da lugar a instrucciones más cortas. Por otro lado, los programas contienen más instrucciones, lo que normalmente supone mayor tiempo de ejecución y programas más largos y más complejos. Con instrucciones de una dirección, el programador tiene generalmente a su disposición solo un registro de uso general (el acumulador). Con instrucciones de múltiples direcciones suele disponerse de múltiples registros de uso general. Esto permite que algunas operaciones se realicen solo con registros. Ya que los accesos a los registros son

más rápidos que a memoria, se acelera la ejecución. Por razones de flexibilidad y facilidad para utilizar varios registros, la mayoría de las máquinas actuales emplean una mezcla de instrucciones de dos y tres direcciones.

Otro aspecto a considerar es si una dirección hace referencia a una posición de memoria o a un registro. ya que hay menos registros, se necesitan menos bits para referenciarlos. Además, como veremos más adelante, una máquina puede permitir diversos modos de direccionamiento y la especificación del modo puede consumir bits adicionales. El resultado es que la mayoría de los diseños de CPU hacen uso de varios formatos de instrucciones.

Diseño del conjunto de instrucciones

Uno de los aspectos más interesantes del diseño de un computador es el diseño del conjunto de instrucciones del lenguaje de máquina. El repertorio de instrucciones define muchas de las funciones realizadas por el CPU y tiene pues un efecto significativo sobre la implementación de la misma. El conjunto de instrucciones es el medio que posee el programador para controlar el CPU. En consecuencia deben considerarse las necesidades del programador a la hora de diseñar el repertorio.

Los aspectos más importantes a considerar son:

- Conjunto de operaciones: cuántas y qué operaciones se van a considerar y cuán complejas serán

- Tipos de datos: los distintos tipos de datos con los que se efectúan las operaciones

- Formatos de instrucciones: longitud de la instrucción en bits, número de direcciones, tamaño de los distintos campos, etc.

- Registros: número de registros del CPU que pueden ser referenciados por las instrucciones y su uso.

- Direccionamiento: modos mediante los cuales puede especificarse la dirección de un operando.

Conjunto de Operaciones

El número de códigos de operación diferentes varía ampliamente de una máquina a otra. Sin embargo, en todas las máquinas podemos encontrar los mismos tipos generales de operaciones. Una clasificación típica es la siguiente:

- Transferencia de Datos
- Aritméticas
- Lógicas
- Conversión
- E/S
- Control de Flujo
- Control del Sistema

Transferencia de Datos

La elección de las instrucciones de transferencia a incluir en un repertorio de instrucciones es un ejemplo del tipo de compromiso a los que debe llegar un

diseñador. Estas instrucciones permuten la copia o duplicación de datos. Se debe especificar la dirección o ubicación del destino y del fuente. Se debe especificar la longitud de los datos a mover (palabra, byte). Afecta los bits de condición según el dato transferido (por ejemplo: N y Z, C=0, V=0).

Por ejemplo, la posición de un operando (memoria o registro) puede indicarse en la especificación del código de la operación o en el operando. Hay diferentes instrucciones para transferencias entre registros, de registro a memoria, de memoria a registro y de memoria a memoria.

move	Transfiere una palabra o un bloque desde una fuente a un destino
store	Transfiere una palabra desde el procesador a memoria
load	Transfiere una palabra desde la memoria al procesador
exchange	Intercambia los contenidos de la fuente y del destino
clear	Transfiere una palabra de ceros al destino
set	Transfiere una palabra de unos al destino
push	Transfiere una palabra desde la fuente a la cabecera de la pila
pop	Transfiere una palabra desde la cabecera de la pila al destino

Aritméticas

La mayoría de las máquinas proporcionan las operaciones aritméticas básicas de suma, resta, multiplicación y división. A veces no se dispone de operaciones de multiplicación, ni división ya que pueden ser implementadas por medio de sumas y restas. Por lo general, se incluyen por razones de

velocidad. Afecta los bits de condición de acuerdo a los resultados obtenidos.

add	Calcula la suma de dos operandos
substract	Calcula la diferencia de dos operandos
multiply	Calcula la multiplicación de dos operandos
divide	Calcula el cociente de dos operandos
absolute	Sustituye el operando por el valor absoluto
negate	Cambia el signo del operando
increment	Suma 1 al operando
decrement	Resta 1 al operando

Lógicas

La mayoría de las máquinas tmabién proporcionan diversas operaciones para manipular bits individuales dentro de una palabra o de una unidad direccionable.

and	Realiza bit a bit la operación lógica and. Puede usarse como máscara para seleccionar ciertos bits de una palabra poniendo en cero los restantes bits. R1=00011010 R2=11110000 R1 and R2 = 00010000
or	Realiza bit a bit la operación lógica or. Puede usarse para encender ciertos bits R1=00011010 R2=10000001 R1 or R2 = 10011011
not	Realiza bit a bit la operación lógica not.

xor	Realiza bit a bit la operación lógica xor. R1=00001111 R2=10101010 R1 xor R2 = 10100101
test	Comprueba la condición especificada, pone los indicadores o banderas en función de los resultados
compare	Realiza la comparación lógica o aritmética de dos o más operandos y pone los indicadores en función del resultado
shift	Desplaza el operando a la izquierda (derecha) el número de bits solicitado. Dos tipos: lógicos y aritméticos R1=10001000 Shiftderecha R1 = 01000100
rotate	Desplaza el operando a la izquierda (derecha) de forma cíclica

El shift aritmético extiende el bit de signo, en tanto que el shift lógico no lo hace.

El shift aritmético afecta los bits de condición N, Z, V y C ($V = N \oplus C$). El shift lógico no afecta el bit de condición V

Transferencia de Control

En los tipos de operación vistos hasta ahora, la instrucción especifica la operación y los operandos. Implícitamente la siguiente instrucción a ejecutar es la inmediatamente posterior, en memoria, de la instrucción en curso. El CPU simplemente incrementa el contador de programas antes de la captación de la siguiente instrucción.

Sin embargo, existen instrucciones que permiten cambiar el flujo de ejecución de un programa. Para estas instrucciones, la operación que realiza el CPU es actualizar el PC para que contenga la dirección de alguna instrucción en memoria.

Las razones por las cuales se necesitan estas operaciones son:

- A veces es esencial poder ejecutar cada instrucción o un conjunto de instrucciones mas de una vez. Importante para la implementación de los lazos.

- Prácticamente todos los programas implican alguna toma de decisión.

- Escribir un programa largo es una tarea excesivamente compleja. Es de gran ayuda particionar un programa en fragmentos más pequeños con los que se trabaje por separado.

jump (branch)	Ruptura incondicional de flujo. Carga el PC con la dirección especificada Jump x PC=x
jump condicional (branch condicional)	Comprueba la condición especificada y dependiendo de la condición carga el PC con la dirección que se especifique
jump to sub	Guarda la información de control del programa en una posición conocida y salta a la dirección especificada
return	Sustituye el contenido del PC y otros registros por los de la posición conocida
execute	Capta el operando de la dirección indicada y lo ejecuta como una instrucción, no modifica el PC
skip	Incrementa el PC

skip condicional	Comprueba la condición indicada y realiza el salto
halt	Detiene la ejecución del programa
wait	Detiene la ejecución del programa, comprueba de forma repetitiva la condición especificada, reanuda la ejecución cuando se satisface la condición
no operation	no se ejecuta operación alguna, pero la ejecución del programa continúa

Una instrucción de bifurcación o de salto tiene como uno de sus operandos la dirección de la siguiente instrucción a ejecutar. Las más frecuentes son las instrucciones de salto condicional. Es decir, se hace la bifurcación sólo si se cumple una condición dada. En caso contrario se ejecuta la siguiente instrucción (se incrementa el contador de programa de la forma habitual).

Hay varias formas de comprobar la condición de una instrucción de salto condicional. En primer lugar, la mayoría de las máquinas proporcionan un código de condición de uno o varios bits que se actualizan cuando se ejecutan algunas operaciones. Por ejemplo, las operaciones aritméticas podrían fijar un código de condición con uno de los cuatros valores siguientes: 0, positivo, negativo, desbordamiento. En tal caso, la máquina podría tener cuatro instrucciones de bifurcación o salto condicional:

BRP X Saltar a la posición X si el resultado es positivo

BRN X Saltar a la posición X si el resultado es negativo

BRZ X Saltar a la posición X si el resultado es cero

BRO X Saltar a la posición X si se ha producido un desbordamiento

En todos los casos anteriores, el resultado al cual se hace referencia es el de la última operación ejecutada que afecte el código de condición.

Otra forma es usando un formato de instrucción de tres direcciones que consiste en realizar la comparación y especificar la bifurcación en la misma instrucción. Por ejemplo:

BRE R1, R2, X Saltar a X si el contenido de R1 es igual al de R2

Otra forma común de instrucciones de control es la instrucción de salto implícito (skip). Esta instrucción incluye una dirección de manera implícita. Dado que esta instrucción no requiere un campo de dirección, éste queda libre para otras cosas. Un ejemplo típico es la instrucción "incrementar y saltar si es cero" ISZ.

Otra instrucción de control importante es aquella que nos permuta llamar a una subrutina. El uso de subrutinas requiere de dos instrucciones básicas: una instrucción de llamada (call) que produce una bifurcación desde la posición actual a la subrutina; y una instrucción de retorno de la subrutina (return) al lugar desde el cual se llamó.

Las subrutinas deben permitir que puedan llamarse desde distintos puntos. Por lo tanto, es necesario que el CPU preserve la dirección de retorno en algún sistio para que pueda realizarse correctamente. Hay tres lugares habituales para guardar la dirección de retorno:

Un registro

Al principio de la subrutina

En la cabecera de la pila

Entrada / Salida

Input	Transfiere datos desde un puerto o dispositivo de E/S al destino (memoria principal o registro)
Output	Transfiere datos desde la fuente especificada a un puerto o dispositivo de E/S
Start I/O	Transfiere instrucciones al procesador de E/S para iniciar operaciones de E/S
Test I/O	Transfiere información de estado desde el sistema de E/S al destino especificado

Conversión

Las instrucciones de conversión son aquellas que cambian el formato. Por ejemplo convertir de entero a punto flotante.