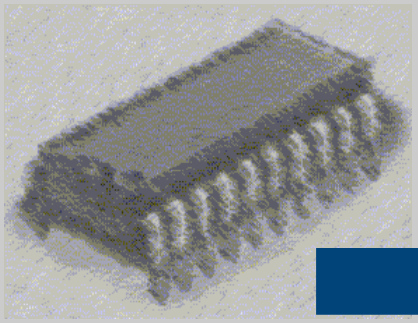
Soluciones a los problemas impares

Tema 2. Diseño del repertorio de instrucciones

Arquitectura de Computadores



I. T. Informática de Gestión

Curso 2009-2010

Base teórica

Al diseñar un computador, uno de los parámetros a tener en cuenta es el repertorio de instrucciones. Dependiendo del tipo de juego deseado, CISC o RISC, de os modos de direccionamiento empleados, del número de operandos con los que trabajan las operaciones de la ALU y del lugar de almacenamiento de los operandos temporales, se verá influido el rendimiento del mismo y la facilidad o dificultad para crear código ejecutable por parte de los compiladores y ensambladores.

Formato de instrucciones

Las instrucciones deben encajarse en unos pocos formatos. Además deben estar formados por campos sistemáticos en aras de facilitar la decodificación de las mismas por parte de la Unidad de Control.

Los computadores emplean un direccionamiento implícito, es decir, la instrucción siguiente a la que se encuentra en curso, es la que ocupa la siguiente posición de memoria, salvo que sea una instrucción de salto condicional, de bifurcación o llamada / regreso de un procedimiento o interrupción. Es por ello, que este tipo de computadores cuenta con un registro contador de programa (CP) que contiene la dirección de memoria en la que se encuentra la instrucción a ejecutar.

Un formato típico de una instrucción se muestra en la figura siguiente:

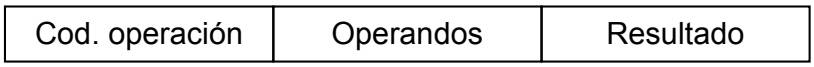


Figura 1. Formato de instrucciones

En muchos computadores comerciales el campo resultado suele eliminarse, de tal modo que uno de los operandos funciona como origen y destino de la operación.

Las instrucciones proporcionan, el tamaño, tipo de operandos y modos de direccionamiento que emplean y suelen ser múltiplos de las palabras del computador.

En algunos repertorios, el código de operación se extiende para hacer que las instrucciones más empleadas tengan un código de operación menor que las menos usadas.

Modos de direccionamiento

Indican la manera de acceder al lugar en el que se encuentran los operandos o en el que se debe almacenar el resultado.

Dependiendo del computador, pueden recibir uno u otro nombre. Los modos de direccionamiento que se pueden encontrar en un computador comercial suelen ser:

- **Inmediato**. El operando se encuentra en la propia instrucción.
- **Directo a registro**. El operando se encuentra en un registro.
- **Directo a memoria**. El operando se encuentra en una posición de memoria.
- **Relativo**. El operando se encuentra en una posición de memoria cuya dirección se calcula realizando la suma del contenido de un registro con un desplazamiento sobre el mismo. Dependiendo del registro empleado existirán diferentes tipos de direccionamiento relativo: a contador de programa, a registro base, mediante registro índice, etc.
- **Indirecto**. La posición de memoria especificada no es el operando sino la dirección de memoria en la que se encuentra.
- **Implícito**. El operando no aparece explícitamente en la instrucción.

Compiladores

Son los encargados de traducir un programa en lenguaje de alto nivel a código ejecutable por el computador. Dependiendo del formato de instrucciones, de los operandos que empleen las operaciones de la ALU, del número de registros de la arquitectura, etc. tendrán una mayor facilidad o dificultad para generar el código.

Las fases de un compilador suele ser la mostrada en la figura siguiente.

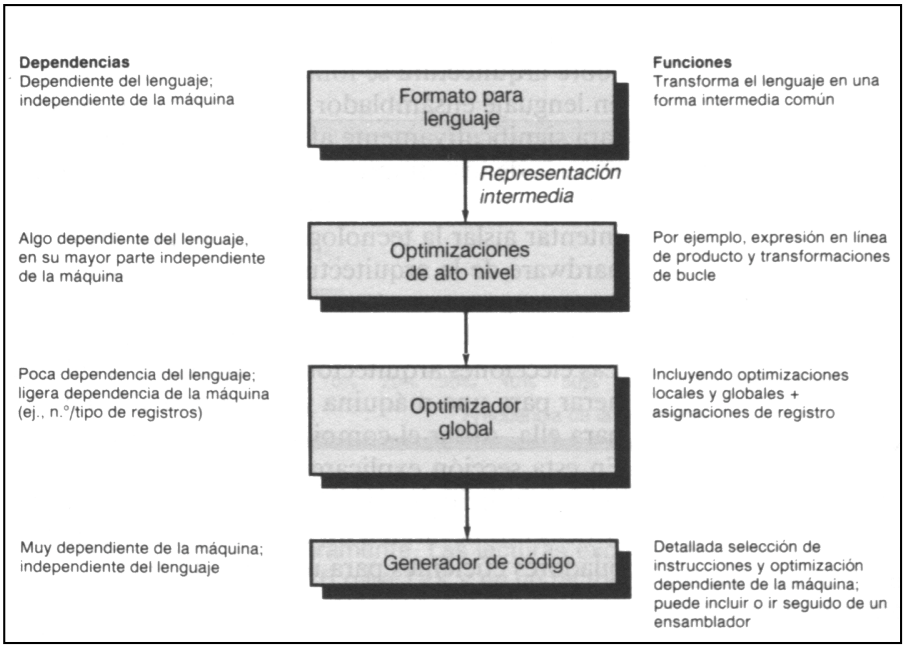


Figura 2. Fases de un compilador

A la hora de tratar el programa fuente y de generar el código el compilador realiza una serie de suposiciones y optimizaciones entre las que destacan:

Nombre de la optimización	Explicación
Integración de procedimientos	Decide si se expanden o no los procedimientos
Eliminación global de subespresiones comunes	Sustituye dos instancias del mismo cálculo por simple copia
Reducción de la altura de la pila	Reorganiza las expresiones matemáticas para minimizar los recursos necesarios para evaluarla
Movimiento de código	Elimina código de un bucle que calcula el mismo valor en cada iteración del bucle
Reducción de potencia	Sustituir la multiplicación por suma s y desplazamientos, la división por restas y desplazamientos
Planificación de la segmentación	Reordenar las instrucciones para mejorar el rendimiento de la segmentación

1. Sea un computador con palabras de 32 bits y 16 registros de 32 bits. De estos registros el .1 es el contador de programa y el .2 el puntero e pila, los demás son de propósito general. La memoria es de 256 Mpalabras

   El juego de instrucciones de esta máquina se reduce a dos instrucciones ortogonales:

   - Move origen, destino
   - Add destino, operando1, operando2

   Los modos de direccionamiento permitidos son: inmediato, directo a registro y a memoria, relativo a registro, a registro índice con pre y pos decremento e incremento y el indirecto.

   El juego de instrucciones se forma con el código de operación, y el campo de cada operando debe llevar asociado su modo de direccionamiento.

   Se pide diseñar los formatos de instrucción de la máquina descrita.

---

2. Sea un computador con 16 Registros, cuya longitud de palabra es de 2 bytes. Diseñar los formatos para las instrucciones de tipo Registro-Registro, utilizando la técnica de “expansión de código de operación” de modo que permita:

   - 15 instrucciones de 3 operandos
   - 14 instrucciones de 2 operandos
   - 31 instrucciones de 1 operando
   - 16 instrucciones de 0 operandos

   Indique en los esquemas las longitudes de todos los campos y, para los códigos de operación también los rangos de valores

---

3. Sea un computador con palabras de 16 bits y 32 registros de 16 bits, que ejecuta el siguiente juego de instrucciones ortogonales:

   - Move fuente, destino
   - Movec fuente, destino, condicion
   - Moved fuente1, destino1, fuente2, destino2, condicion
   - Add operando1, operando2, destino
   - Sub operando1, operando2, destino
   - Mul operando1, operando2, destino
   - Div operando1, operando2, destino
   - And operando1, operando2, destino
   - Or operando1, operando2, destino
   - Xor operando1, operando2, destino
   - Shift fuente, destino, tipo, contador

   Donde:

   - La condición puede ser C, NC, Z y NZ
   - Los modos de direccionamiento: inmediato, directo a registro y relativo a registro
   - Los datos pueden ser enteros sin signo, enteros en complemento a 2 y reales en coma flotante
   - Cada instrucción solamente opera con todos sus datos en el mismo formato de representación

   Se pide diseñar el formato de las instrucciones para dicho juego de operaciones puras

4. Sea un computador de 32 bits de ancho de palabra y que cuenta con 32 registros de 32 bits.

   Tiene un juego de 60 instrucciones y los modos de direccionamiento:

   - Directo
   - Indirecto
   - A registro base
   - A contador de programa.

   Diseñar el formato de instrucción para las operaciones Reg-Mem

5. Diseñar un código de operación extendida que permita codificar en una instrucción de 36 bits la siguiente información:

   - 7 instrucciones con dos direcciones de 15 bits y una dirección de 3 bits
   - 500 instrucciones de una dirección de 15 bits y una de 3 bits
   - 50 instrucciones de 0 direcciones

6. Sea un computador con 16 Registros, cuya longitud de palabra es de 2 bytes. Diseñar los formatos para las instrucciones de tipo Registro-Registro, utilizando la técnica de “expansión de código de operación” de modoque permita:

   - 15 instrucciones de 3 operandos
   - 14 instrucciones de 2 operandos
   - 31 instrucciones de 1 operando
   - 16 instrucciones de 0 operandos

Indique en los esquemas las longitudes de todos los campos y, para los códigos de operación también los rangos de valores.

7. Suponiendo que el 90 % de los saltos hacia atrás son efectivos. Calcular la probabilidad de que un salto hacia delante sea efectivo empleando la media de los datos de la tabla siguiente

Programa	Porcentaje de saltos hacia atrás	Porcentaje de saltos efectivos	Porcentaje de todas las instrucciones de control que realmente saltan
TeX	17%	54%	70%
Spice	31%	51%	63%
GCC	26%	54%	63%

8. Se tiene el mismo juego de instrucciones implementado en dos computadores con la misma arquitectura. Las características de cada una al ejecutar el mismo programa se resumen en la tabla siguiente:

	Ciclo de reloj	Ciclos por instrucción (CPI) para el programa
Arquitectura 1	3	2
Arquitectura 2	4	1,5

Se pide calcular qué máquina es más rápida para ese programa y cuánto más

9. Se tiene el mismo juego de instrucciones implementado en dos computadores con la misma arquitectura. Las características de cada una al ejecutar el mismo programa se resumen en la tabla siguiente:

	Ciclo de reloj	Ciclos por instrucción (CPI) para el programa
Arquitectura 1	2	2
Arquitectura 2	4	1,2

Se pide calcular qué máquina es más rápida para ese programa y cuánto más

10. Sea un computador con 8 registros, cuya longitud de palabra es de 2 bytes. Diseñar los formatos para las instrucciones de tipo Registro-Registro, utilizando la técnica de "expansión de código de operación" de modo que permita:
    - 120 instrucciones de 3 operandos
    - 5 instrucciones de 2 operandos
    - 12 instrucciones de 1 operando
    - 7 instrucciones de 0 operandos

11. Sea un computador con 8 registros, cuya longitud de palabra es de 2 bytes. Diseñar los formatos para las instrucciones de tipo Registro-Registro, utilizando la técnica de "expansión de código de operación" de modo que permita:
    - 127 instrucciones de 3 operandos
    - 6 instrucciones de 2 operandos
    - 15 instrucciones de 1 operando
    - 8 instrucciones de 0 operandos

Solución ejercicio 1

Como cada operando puede tener cualquier modo de direccionamiento, la máquina tendrá un formato por modo de direccionamiento diferente. Con lo que para distinguir los 8 modos de direccionamiento se requerirán 3 bits.

Al contar únicamente con dos instrucciones, bastará un bit para diferenciarlas.

Los 16 registros necesitaran 4 bits para ser diferenciados. Finalmente, para poder direccionar 256 Mpalabras se necesitarán 28 bits (256 Mp. = 2^{28})

Una solución posible será:

Para la instrucción MOVE origen, destino

Cod. Operación	M.D.1	Dato	No usado	M.D.2.	Dato
(1)	(3)	(28)	(1)	(3)	(28)

Para la instrucción ADD destino, operando1, operando2

Cod. Op.	MD1	dato	No usado	MD2	dato	No usado	MD3	Dato
(1)	(3)	(28)	(1)	(3)	(28)	(1)	(3)	(28)

La pareja, MDn + dato variará según los modos de direccionamiento, siendo:

Para los modos de direccionamiento inmediato, directo a memoria e indirecto

Modo de direccionamiento	Dirección de memoria o inmediato
(3)	(28)

Y para el resto de los modos de direccionamiento:

Formato para el direccionamiento directo a registro:

Modo de direc.	Registro	No usado
(3)	(4)	(24)

Formato para el direccionamiento relativo a registro:

Modo de direc.	Registro	Desplazamiento
(3)	(4)	(24)

Formato para el direccionamiento relativo a registro índice con pre (pos) incremento (decremento):

Modo de direc.	Registro	Inc / Dec	Pre / Pos	Desplazamiento
(3)	(4)	(1)	(1)	(22)

Solución ejercicio 3

Al ser un juego de instrucciones puras, el código de operación no debe contener ninguna información sobre el modo de direccionamiento de los operandos, ni sobre el formato de representación de los mismos ya que cualquier instrucción deberá poderse realizar con cualquier modo de direccionamiento y con cualquier tipo de datos.

Dado que las instrucciones operan con el mismo tipo de datos, el campo puede ser común pero no así el modo de direccionamiento que puede ser diferente para los operandos

De esta forma se necesitarán:

Bits necesarios	Distingue entre
4	11 instrucciones diferentes
2	3 modos de direccionamiento distintos
2	3 sistemas de representación diferentes
5	32 registros
2	4 condiciones

MOVE fuente, destino

CO	SRD	MD1	MD2	No usado	Fuente	Destino
(4)	(2)	(2)	(2)	(6)	(16)	(16)

MOVEC fuente, destino, condición

CO	SRD	MD1	MD2	N/U	COND	Fuente	Destino
(4)	(2)	(2)	(2)	(4)	(2)	(16)	(16)

MOVED fuente1, destino1, fuente2, destino,2 condición

CO	SRD	MD1	MD2	MD3	MD4	COND	dato	dato	dato	dato
(4)	(2)	(2)	(2)	(2)	(2)	(2)	(16)	(16)	(16)	(16)

Aritméticas y lógicas

CO	SRD	MD1	MD2	MD3	N/U	dato	dato	dato
(4)	(2)	(2)	(2)	(2)	(4)			

Shift fuente, destino, tipo, contador

CO	SRD	MD1	MD2	TP	CONT	dato	dato
(4)	(2)	(2)	(2)	(2)	(4)	(16)	(16)

Aunque todos los datos ocupan 16 bits, su formato interno dependerá de su modo de direccionamiento

Inmediato los 16 bits estarán en uno de los posibles sistemas de representación: entero sin signo, complemento a dos o coma flotante

Directo a registro

Registro	No usado
(5)	(11)

Relativo el campo Desplazamiento no se usará en el modo directo a registro

Registro	Desplazamiento
(5)	(11)

Solución ejercicio 5

Formato de 7 instrucciones con tres direcciones dos de 15 bits y una de 3 bits.

Para codificar 7 instrucciones necesitaremos 3 bits y sobrará una combinación, con lo que el formato será:

Código operación	Dirección1	Dirección2	Dirección3
(3)	(15)	(15)	(3)

Formato de 500 instrucciones con dos direcciones una de 15 bits y otra de 3 bits

Se necesitarán al menos 9 bits para codificarlas (sobrarán 12 combinaciones) Para poder distinguir estas instrucciones de las anteriores podemos poner los 3 primeros bits del código de operación a 111 (antes sobraba una combinación) con lo que el código de operación serán 12 bits. El formato quedará

1 1 1	Código operación	No usado	Dirección1	Dirección2
(3)	(9)	(6)	(15)	(3)

Formato de las cincuenta instrucciones de cero direcciones

Las 50 instrucciones las distinguiremos de las anteriores poniendo los 12 primeros bits a uno. El código de operación batará con 6 bits (64 combinaciones)

1 1 1	1 1 1 1 1 1 1 1 1	Código operación	No usado
(3)	(9)	(6)	(18)

Solución ejercicio 7

Calculando la media para los tres programas nos da:

- Porcentaje de saltos hacia atrás: 25%
- Porcentaje de saltos efectivos: 53%
- Porcentaje total de bifurcaciones: 65%

La frecuencia media de los saltos efectivos será la suma de las frecuencias de los efectivos hacia detrás y hacia delante

%efectivos=(%efectivos hacia atrás x %hacia atrás)+(%efectivos hacia delante x %hacia delante)

53 % = (90% x 25 %) + (%efectivos hacia delante x 75%)

De donde %efectivos hacia delante =(53% - 22,5%) / 75 = 40,7%

Solución ejercicio 9

Como se trata del mismo programa el número de instrucciones será el mismo en ambas máquinas. Con lo que

- Ciclos de reloj CPUA = Número instrucciones x 2
- Ciclos de reloj CPUB = Número instrucciones x 1,2

El tiempo de ejecución de CPU para cada máquina será:

- Tiempo CPUA = Ciclos de reloj CPUA x tiempo de cicloA = Núm. Instruc. x 2 x 2ns
- Tiempo CPUB= Ciclos de reloj CPUB x tiempo de cicloB = Núm. Instruc. x 1,2 x 4ns

Como puede observarse en la tabla siguiente la máquina A es más rápida que la B

	Ciclos de Reloj	Tiempo de CPU
Arquitectura A	2 x Núm. de instruc.	4 x Núm. de instruc.
Arquitectura B	1,2 x Núm. de instruc.	4,8 x Núm. de instruc.

Finalmente se ve que la máquina A es 1,2 veces más rápida que la B

$$\frac{\text{Rendimiento CPU}_A}{\text{Rendimiento CPU}_B} = \frac{\text{Tiempo de ejecución}_B}{\text{Tiempo de ejecución}_A} = \frac{4{,}8 x Número de Instrucciones}{4 x Número de Instruciones} = 1{,}2 veces$$

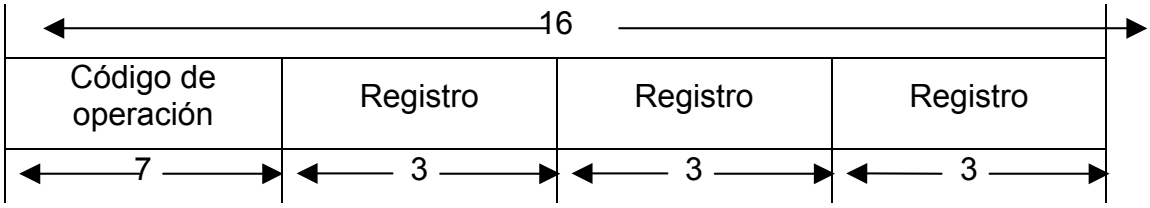
Solución ejercicio 11

Se tienen palabras de 2 bytes con lo que contamos con 16 bits para codificar el formato de instrucción.

Tenemos 8 registros, con lo que para diferenciarlos necesitaremos 3 bits.

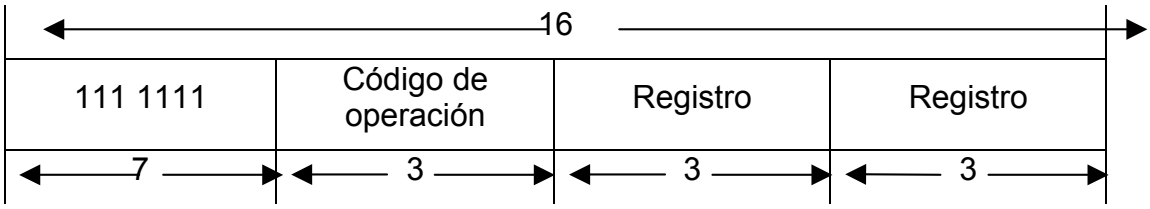
Al pedirnos que se realice con la técnica de expansión de código, deberemos comprobar que quedan combinaciones libres para distinguir unos formatos de otros

127 instrucciones con tres operandos



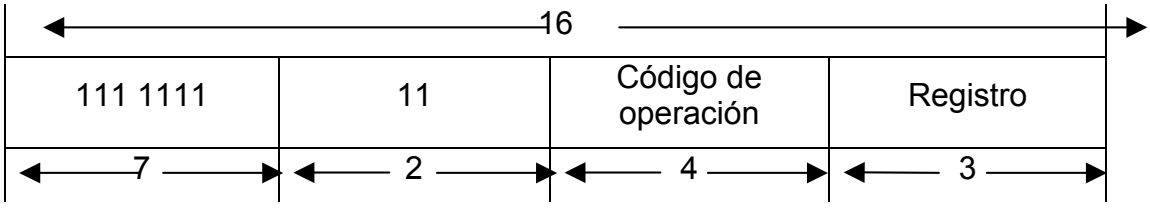
Con 7 bits contamos con 128 combinaciones, como solamente empleamos 127 nos sobra una para distinguir entre este formato y el siguiente

6 instrucciones con 2 operandos



Con 3 bits tenemos 8 combinaciones, nos quedan dos libres, por ejemplo: 110 y 111.

15 instrucciones con 0 operando



Para 15 combinaciones necesitamos 4 bits. Por lo que nos aprovechamos de que antes nos sobraban dos combinaciones que empiezan por 11

7 instrucciones con cero operandos