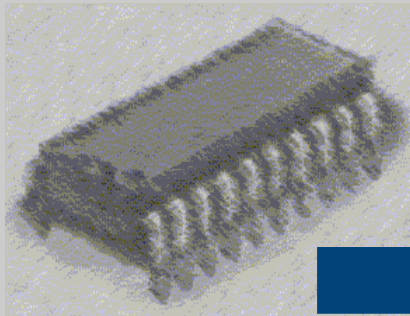


*Enunciados de problemas*

## **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 los 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:

Cod. operación	Operandos	Resultado
----------------	-----------	-----------

*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.

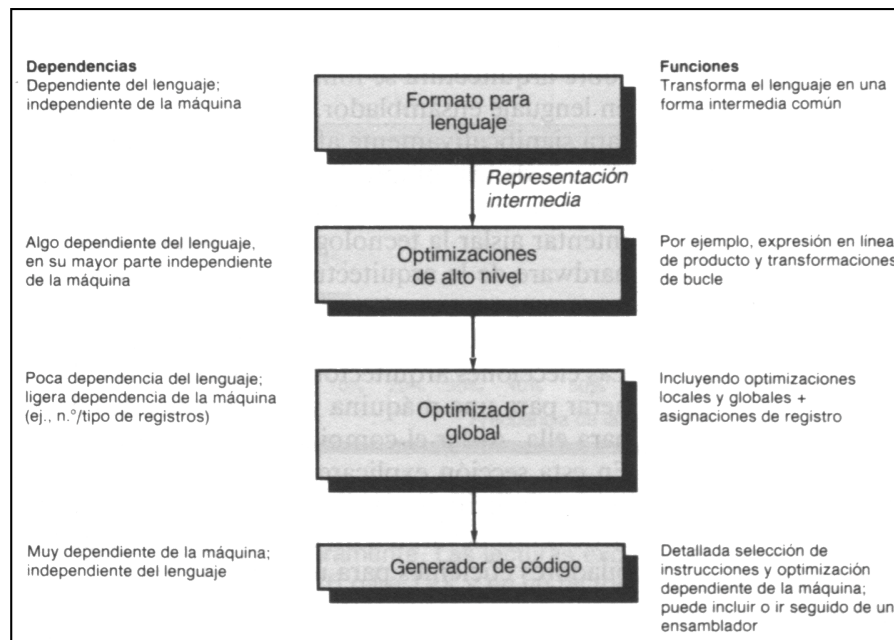


Figura 2. Fases de un compilador

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 subexpresiones 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 sumas 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 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.

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
<i>Arquitectura 2</i>	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
<i>Arquitectura 2</i>	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