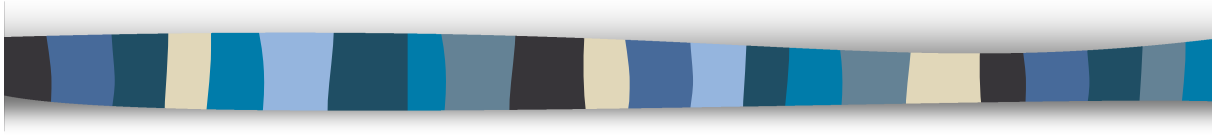


Formato de instrucciones



Material Elaborado por el Profesor Ricardo González
A partir de Materiales de las Profesoras
Angela Di Serio
María Blanca Ibañez

1

Elementos de una instrucción de máquina

add rd rs rt

Función: **rd** ← **rs+rt**

add \$8, \$2, \$3 => 00434020

Cód. Op.	Registro fuente 1	Registro fuente 2	Registro destino		Funct
0	rs	rt	rd	0	0x20
6	5	5	5	5	6

000000 00010 00011 01000 00000 100000

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 esta 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, se debe suministrar la dirección de memoria.

2

Como se analiza una instrucción en el CPU

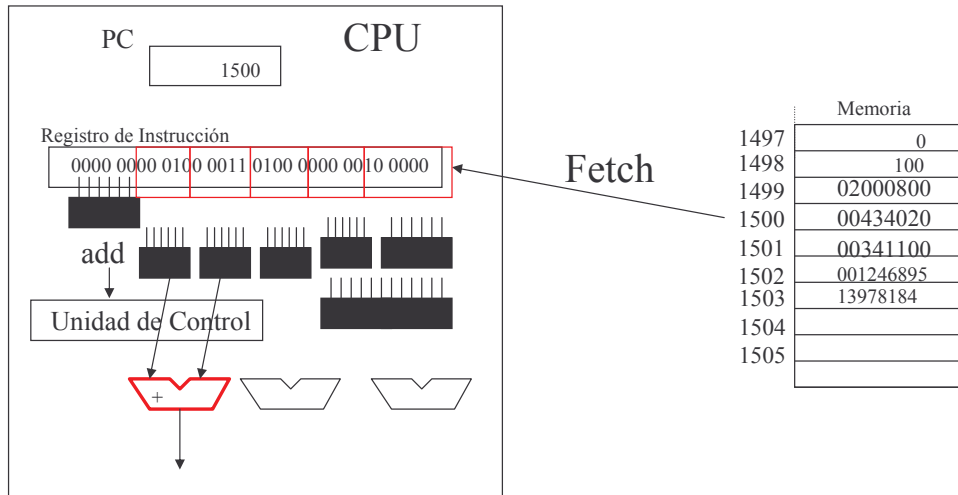
add rd rs rt

Función: $rd \leftarrow rs + rt$

add \$8, \$2, \$3 => 00434020

Cód. Op.	Registro fuente 1	Registro fuente 2	Registro destino	Funct	
0	rs	rt	rd	0	0x20
6	5	5	5	5	6

000000 00010 00011 01000 00000 100000
 0 | 0 | 4 | 3 | 4 | 0 | 2 | 0



3

Formato de instrucciones

- El formato de las instrucciones es un conjunto de especificaciones que indican como debe ser interpretado el patrón de bits de una instrucción de máquina para lograr su ejecución dentro del computador.
- El formato de la instrucción nos indica cual es el código de operación y cuales los operandos que la instrucción especifica, tanto explícita como implícitamente.

4



Criterios para el diseño de los formatos de las instrucciones

- 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 su 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.

5



Criterios para el diseño de los formatos de las instrucciones

- 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 diseño de ese 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 parezcan luego que fueron las mejores.

6



Criterios para el diseño de los formatos de las instrucciones

- 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.
- Si se minimiza mucho el tamaño de las instrucciones puede hacer que el proceso de decodificación y sobreposición de instrucciones sea más complicado.

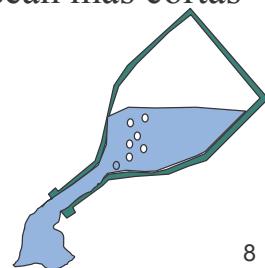
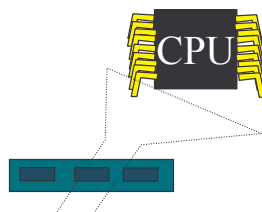


7



Criterios para el diseño de los formatos de las instrucciones

- Un motivo para disminuir el tamaño de las instrucciones es el ancho de banda de la memoria, es decir, el número de bits 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.



8



Criterios para el diseño de los formatos de las instrucciones

- 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.

9



Criterios para el diseño de los formatos de las instrucciones

- Existe un compromiso entre el deseo de disponer de 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.¹⁰

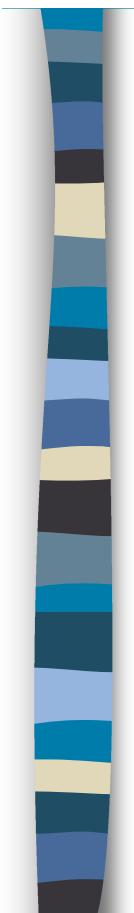


Criterios para el diseño de los formatos de las instrucciones

+	0	+	00
*	1	-	01
		*	10
		/	11

- Otro criterio de diseño que debe tomarse en cuenta es 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.

11



Diseño de los formatos de las instrucciones

Teniendo en cuenta estos criterios de diseño, vamos a estudiar en que 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 implícitamente o explícitamente el modo de direccionamiento para cada operando.
- Por lo general se emplea 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 variable.

12



Diseño de los formatos de las instrucciones

- Debe existir un compromiso entre el número de códigos de operación a emplear vs. capacidad de direccionamiento.



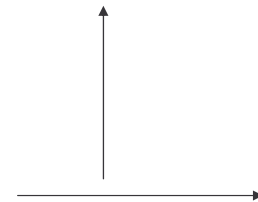
En referencia a la definición de los operandos se deben considerar los siguientes factores:

- El número de modos de direccionamiento.
- El número de operandos.
- El acceso a Registros frente al acceso a Memoria
- Rango de direcciones

13



Diseño de los formatos de las instrucciones



- Hay conjuntos de instrucciones que presentan la propiedad de ser ortogonales. En el caso de formato de instrucciones, éste 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.
- Hay arquitecturas que trabajan con instrucciones con una única longitud (fija) y hay otras que adoptan formatos con distintas longitudes.
- El direccionamiento puede ser más flexible con varias combinaciones de referencias a registros y a memoria, así como de modos de direccionamiento.

14

Formato de instrucciones de MIPS.

Format R	R	op	rs	rt	rd	shamt	funct	Formato de instrucciones aritméticas
Format I	I	op	rs	rt	dirección/inmediata			Formato de transferencia, salto, inmediato
Format J	J	op	target address					Formato de instrucciones de bifurcación
Field size		6 bits	5 bits	5 bits	5 bits	5 bits	5 bits	Todas instrucciones MIPS de 32 bits

- Los formatos ejemplos se muestran con valores en cada campo: los campos de **op** y **funct** forman el código de operación (cada uno 6 bits), el campo **rs** da un registro fuente (5 bits), **rt** es también normalmente un registro fuente (5 bits), **rd** es el registro destino (5 bits) **shamt** da la cantidad de desplazamiento.

15

Formato de instrucciones de MIPS.

Nombre	Formato	6 bits	5 bits	Ejemplo		5 bits	6 bits	Comentarios
add	R	0	2	3	1	0	32	add \$1,\$2,\$3
sub	R	0	2	3	1	0	34	sub \$1,\$2,\$3
addi	I	8	2	1		100		addi \$1,\$2,100
addu	R	0	2	3	1	0	39	addu \$1,\$2,\$3
subu	R	0	2	3	1	0	35	subu \$1,\$2,\$3
addiu	I	9	2	1		100		addiu \$1,\$2,100
mfc0	R	18	0	1	14	0	0	mfc0 \$1,\$cpc
mult	R	0	2	3	0	0	24	mult \$2,\$3
multu	R	0	2	3	0	0	25	multu \$2,\$3
div	R	0	2	3	0	0	26	div \$2,\$3
divu	R	0	2	3	0	0	27	divu \$2,\$3
mflr	R	0	0	0	1	0	16	mflr \$1
mflo	R	0	0	0	1	0	18	mflo \$1
and	R	0	2	3	1	0	36	and \$1,\$2,\$3
or	R	0	2	3	1	0	37	or \$1,\$2,\$3
andi	I	12	2	1		100		andi \$1,\$2,100
ori	I	13	2	1		100		ori \$1,\$2,100
sll	R	0	0	2	1	10	0	sll \$1,\$2,10
srl	R	0	0	2	1	10	2	srl \$1,\$2,10
lwr	I	35	2	1		100		lwr \$1,100(\$2)
sw	I	43	2	1		100		sw \$1,100(\$2)
lui	I	15	0	1		100		lui \$1,100
beq	I	4	1	2		100		beq \$1,\$2,100
bne	I	5	1	2		100		bne \$1,\$2,100
slt	R	0	2	3	1	0	42	slt \$1,\$2,\$3
slti	I	10	2	1		100		slt \$1,\$2,100
sltu	R	0	2	3	1	0	43	sltu \$1,\$2,\$3
sltiu	I	11	2	1		100		sltiu \$1,\$2,100
j	J	2			10000			j 10000
jr	R	0	31	0	0	0	8	jr \$1
jal	J	3			10000			jal 10000

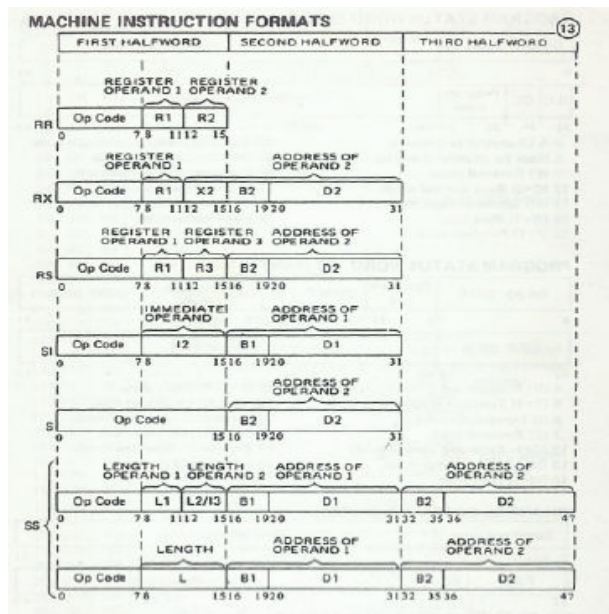
16

Formatos de instruccion usados en el Motorola 68000 (solo primera palabra).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	COP		TAM		OPERANDO				OPERANDO				MOVE				
2	COP		REG		MOD		OPERANDO				ADD, AND, CHP, SUB						
3	COP		REG		COP		OPERANDO				CHK, DIVS, LEA, MULS						
4	COP		REG		MOD		COP		REG		MOVEP						
5	COP		REG		COP		TAM		COP		REG		ASL, ASR, ROL, ROR				
6	COP		REG		COP				REG				ABCD, EXG, SBGD				
7	COP		REG		COP		DATA				MOVEQ						
8	COP		CUENTA		COP		TAM		COP		REG		ASL, ASR, ROL, ROR				
9	COP		DATO		COP		TAM		OPERANDO				ADDQ, SUBQ				
10	COP		CONDICION				COP		OPERANDO				ScC				
11	COP		CONDICION				DESPLAZAMIENTO				Bcc						
12	COP		CONDICION				COP		REG				DBcc				
13	COP				TAM		OPERANDO				ADDI, CHIPL, NEG, TST						
14	COP				TAM		OPERANDO				MOVEM						
15	COP				OPERANDO				JMP, JSR, NBCD, PEA								
16	COP				VECTOR				TRAP								
17	COP				REG				EXT, LINK, SWAP, UNLINK								
18	COP								NOP, RESET, RTS, TRAPV								

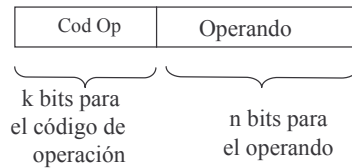
COP determina la instruccion
 OPERANDO determina un dato sobre el que se opera.
 REG selecciona un registro
 TAM escoge entre byte, palabra y palabra larga.
 MOD determina si OPERANDO es fuente o destino y su longitud
 CUENTE y DATA son constantes
 CONDICION especifica una de las 16 condiciones posibles a examinar
 DESPLAZAMIENTO es un desplazamiento con signo para las bifurcaciones
 VECTOR especifica donde hacer un desvio

Formatos de instruccion usados en las arquitecturas IBM 360/370/390/ Serie Z





Expansión de códigos de operación



- Considere una instrucción de $(k+n)$ 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 $(k+n)$ bits 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.

19



Expansión de códigos de operación

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:

4 bits => 16 valores

- 15 instrucciones de 3 direcciones

$$15 + 14 + 31 + 16 = 76$$

- 14 instrucciones de dos direcciones

$$76 > 16$$

- 31 instrucciones de una dirección

- 16 instrucciones de cero direcciones

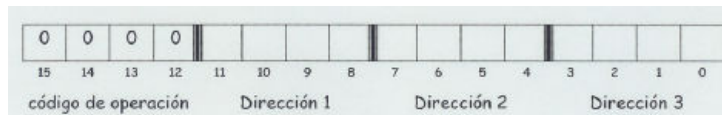
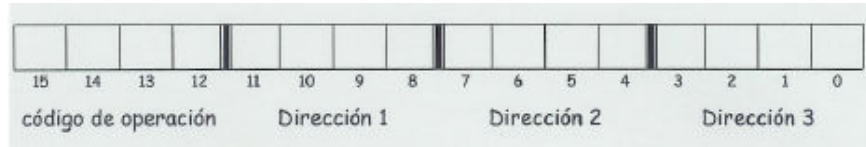


20

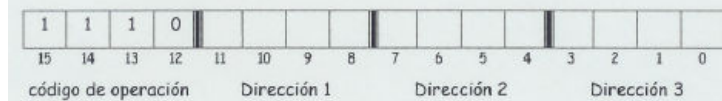
Expansión de códigos de operación

15 instrucciones de 3 direcciones

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.



...



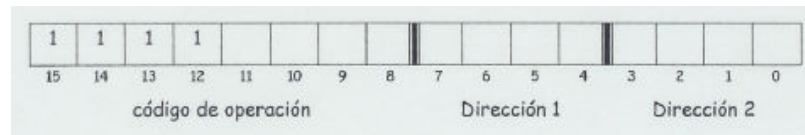
0000 1
0001 2
0010 3
0011 4
0100 5
0101 6
0110 7
0111 8
1000 9
1001 10
1010 11
1011 12
1100 13
1101 14
1110 15

1111 16
21

Expansión de códigos de operación

14 instrucciones de dos direcciones

- El código 15 indica que el código de operación está contenido realmente en los bits del 8 al 15.



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 del 8 al 11 tendremos las distintas combinaciones para nuevos códigos de operación.

1111 0000 1
1111 0001 2
1111 0010 3
1111 0011 4
1111 0100 5
1111 0101 6
1111 0110 7
1111 0111 8
1111 1000 9
1111 1001 10
1111 1010 11
1111 1011 12
1111 1100 13
1111 1101 14

1111 1110 15
1111 1111 16
22

Expansión de códigos de operación

Rango de Códigos de operación para las 14 instrucciones de dos direcciones

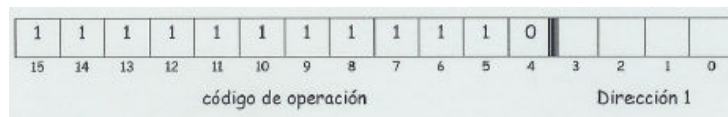
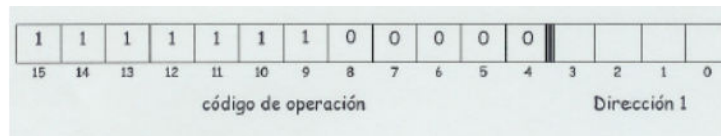


Las instrucciones que tienen 1111 en los bits más a la izquierda y 1110 y 1111 en los bits 8 a 11 serán tratadas de forma distinta.

Expansión de códigos de operación

31 instrucciones de una dirección

Estas instrucciones se tratarán como si el código de operación está en los bits 4 a 15



- 1111 1110 0000 1
- 1111 1110 0001 2
- 1111 1110 0010 3
- 1111 1110 0011 4
- 1111 1110 0100 5
- 1111 1110 0101 6
- 1111 1110 0110 7
- 1111 1110 0111 8
- 1111 1110 1000 9
- 1111 1110 1001 10
- 1111 1110 1010 11
- 1111 1110 1011 12
- 1111 1110 1100 13
- 1111 1110 1101 14
- 1111 1110 1110 15
- 1111 1110 1111 16

- 1111 1111 0000 17
- 1111 1111 0001 18
- 1111 1111 0010 19
- 1111 1111 0011 20
- 1111 1111 0100 21
- 1111 1111 0101 22
- 1111 1111 0110 23
- 1111 1111 0111 24
- 1111 1111 1000 25
- 1111 1111 1001 26
- 1111 1111 1010 27
- 1111 1111 1011 28
- 1111 1111 1100 29
- 1111 1111 1101 30
- 1111 1111 1110 31

1111 1111 1111

Expansión de códigos de operación

16 instrucciones de cero direcciones



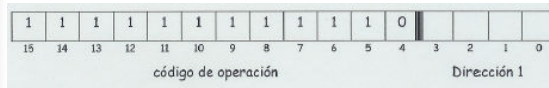
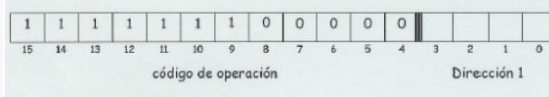
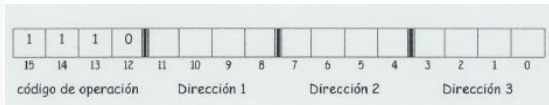
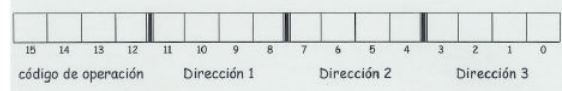
El código de operación 1111 1111 1111 se interpreta como señal de que el código de operación esta en los bits 0 a 15, lo que da 16 instrucciones con cero direcciones.

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, las de una dirección tienen un código de 12 bits y las de cero tienen códigos de 16 bits.

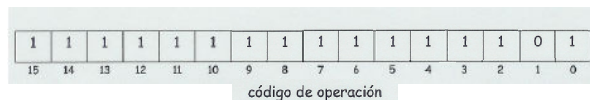
- 1111 1111 1111 0000 1
- 1111 1111 1111 0001 2
- 1111 1111 1111 0010 3
- 1111 1111 1111 0011 4
- 1111 1111 1111 0100 5
- 1111 1111 1111 0101 6
- 1111 1111 1111 0110 7
- 1111 1111 1111 0111 8
- 1111 1111 1111 1000 9
- 1111 1111 1111 1001 10
- 1111 1111 1111 1010 11
- 1111 1111 1111 1011 12
- 1111 1111 1111 1100 13
- 1111 1111 1111 1101 14
- 1111 1111 1111 1110 15
- 1111 1111 1111 1111 16

Expansión de códigos de operación (resumen)

Formato Inicial



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





Expansión de códigos de operación

Con el uso de expansión de códigos de operación se puede minimizar la longitud promedio de las instrucciones codificando cada instrucción de forma 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.