EL CPU-Conjunto de Instrucciones

- Tipos de Operandos
- Tipos de Operadores
- Lenguaje Ensamblador
- Direccionamiento
- Formatos de Instrucciones

Introducción

Se revisa de la unidad aritmético-lógica (ALU) la parte más compleja que es la aritmética del computador.

La aritmética se realiza con dos tipos de números diferentes: *enteros* y *de punto flotante*. La representación elegida es el primer aspecto a tratar y después las operaciones aritméticas.

Tipos de operandos

El diseñador de la computadora y el programador pueden ver la misma máquina desde el repertorio de instrucciones, ya que para el diseñador, el conjunto de instrucciones máquina informa de los requisitos funcionales de la CPU, y el usuario que elige programar en lenguaje máquina tiene que conocer la estructura de registros y de memoria, de los tipos de datos y la funcionalidad de la ALU.

Características de las Instrucciones Máquina

El funcionamiento del CPU está determinado por las instrucciones que ejecuta, conocidas también como *instrucciones máquina* o *instrucciones del computador*. Al conjunto de instrucciones distintas que puede ejecutar el CPU se denomina *repertorio de instrucciones*

Elementos de una instrucción máquina

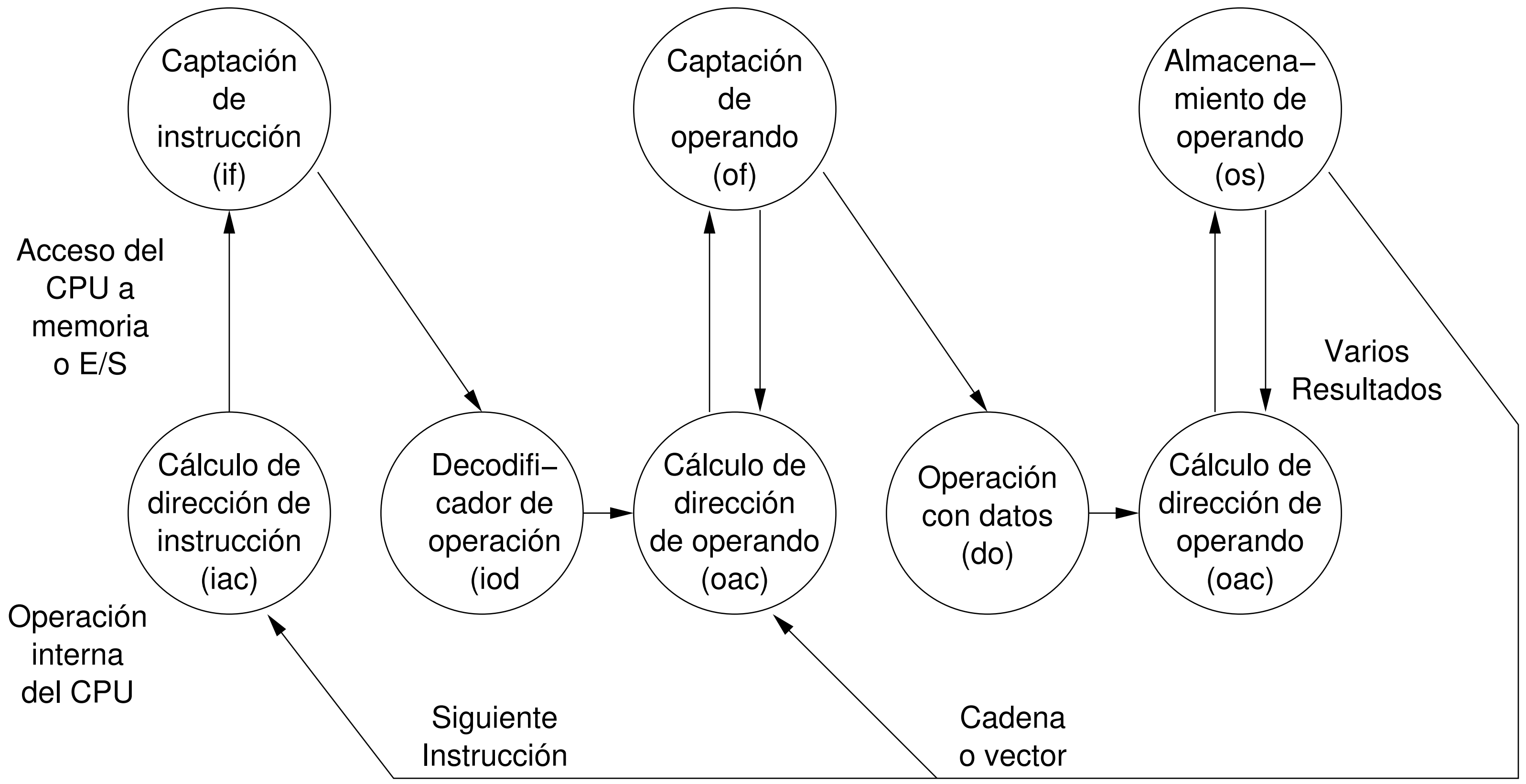
Cada instrucción debe contener la información que necesita el CPU para su ejecución. Los elementos que la constituyen son:

- *Código de operación*
- *Referencia a operandos fuente*
- *Referencia al operando resultado*
- *Referencia a la siguiente instrucción*

Los operandos pueden estar en:

- *Memoria principal o virtual*
- *Registro del CPU*
- *Dispositivo de E/S*

Diagrama de estados de un ciclo de instrucción



Representación de las instrucciones

Cada instrucción se representa por una secuencia de bits. La instrucción está dividida en campos. La descripción se denomina *formato de instrucción*. En la mayoría de los repertorios se emplea más de un formato.

Como sea difícil para los programadores manejar las representaciones binarias de las instrucciones máquina, se usan *representaciones simbólicas*.

Los codops se representan mediante abreviaturas, denominadas *nemotécnicos*.

4	6	6
Codop	Referencia a operando	Referencia a operando

16 bits

Tipos de instrucciones

Cualquier programa escrito con un lenguaje de alto nivel debe traducirse a lenguaje máquina para ser ejecutado, por lo que el repertorio de instrucciones debe ser suficientemente amplio para expresar cualquiera de las instrucciones de alto nivel. Los tipos de instrucciones pueden ser clasificadas como de:

- *Procesamiento de datos.* Aritméticas y lógicas.
- *Almacenamiento de datos.* Instrucciones de memoria.
- *Trasferencia de datos.* Instrucciones de E/S.
- *Control.* Instrucciones de comprobación y de bifurcación.

Número de direcciones

Una de las formas tradicionales de describir la arquitectura de un procesador es en términos del número de direcciones contenidas en cada instrucción. La mayoría de los CPU trabajan con instrucciones con una, dos o tres direcciones.

La instrucción de una sola dirección para que se pueda usar con un operador binario requiere de una dirección implícita, que es un registro del CPU conocido como *acumulador*.

Es posible para algunas de las instrucciones usar cero direcciones y son aplicables a una organización especial de memoria, denominada *pila (stack)*.

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, y se obtienen instrucciones más cortas, pero los programas contienen más instrucciones, con lo que se tiene mayor tiempo de ejecución y programas más largos y complejos. Con instrucciones de múltiples direcciones suele disponerse de múltiples registros de uso general, lo que permite que algunas operaciones se realicen solo con registros, lo que acelera la ejecución. La mayoría de las máquinas actuales usan una mezcla de instrucciones de dos y tres direcciones.

Programas para calcular $Y = (A - B) \div (C + D \times E)$

Instrucción		*Comentario*
SUB	Y,A,B	Y $\leftarrow$ A $-$ B
MPY	T,D,E	T $\leftarrow$ D $\times$ E
ADD	T,T,C	T $\leftarrow$ T $+$ C
DIV	Y,Y,T	Y $\leftarrow$ Y $\div$ T

Instrucciones de tres direcciones

Programas para calcular $Y = (A - B) \div (C + D \times E)$

Instrucción		*Comentario*
MOVE	Y,A	Y $\leftarrow$ A
SUB	Y,B	Y $\leftarrow$ Y $-$ B
MOVE	T,D	T $\leftarrow$ D
MPY	T,E	T $\leftarrow$ T $\times$ E
ADD	T,C	T $\leftarrow$ T $+$ C
DIV	Y,T	Y $\leftarrow$ Y $\div$ T

Instrucciones de dos direcciones

Programas para calcular $Y = (A - B) \div (C + D \times E)$

Instrucción		*Comentario*
LOAD	D	AC $\leftarrow$ D
MPY	E	AC $\leftarrow$ AC $\times$ E
ADD	C	AC $\leftarrow$ AC $+$ C
STOR	Y	Y $\leftarrow$ AC
LOAD	A	AC $\leftarrow$ A
SUB	B	AC $\leftarrow$ AC $-$ B
DIV	Y	AC $\leftarrow$ AC $\div$ Y
STOR	Y	Y $\leftarrow$ AC

Instrucciones de una dirección

Número de Direcciones	*Representación Simbólica*	*Interpretación´*
3	OP A, B, C	$A \leftarrow B\text{OP}C$
2	OP A, B	$A \leftarrow A\text{OP}B$
1	OP A	$AC \leftarrow AC\text{OP}A$
0	OP	$T \leftarrow T\text{OP}(T-1)$

Tipos de Operandos

Las categorias generales más importantes de datos son:

- Direcciones
- Números
- Caracteres
- Datos Lógicos

Las direcciones son un tipo de datos, con las cuales se realizan algunos cálculos para determinar la dirección una dirección de memoria principal o virtual. Por lo que las direcciones pueden considerarse como números enteros sin signo.

Números

Todos los lenguajes máquina incluyen tipos de datos numéricos. Incluso en el procesamiento de datos no numéricos se necesitan números para usarse como contadores longitudes de campo, etc. Los números almacenados en la computadora están limitados, ya que hay un limite para la magnitud de los números y para los números de punto flotante existe un limite de precisión. El programador debe estar consciente de las consecuencias del redondeo, el desbordamiento, o el desbordamiento a cero.

Los tipos de datos numéricos usuales son:

- Enteros o de punto fijo.
- De punto flotante.
- Decimal.

Como los usuarios del sistema usen un sistema decimal, es necesario convertir de decimal a binario las entradas y de binario a decimal las salidas. En aplicaciones donde hay muchas entradas/salidas contra pocos cálculos, y cálculos simples, es preferible guardar y operar con números en su forma decimal. La representación más común es de decimal empaquetado. Muchas máquinas tienen instrucciones aritméticas para operar directamente con números en decimal empaquetado.

Caracteres

Una forma común de datos es el texto o secuencia de caracteres. Los diversos códigos permiten representar caracteres mediante secuencias de bits. El código más utilizado es el ASCII. Otro código utilizado es el EBCDID (Extended Bynary Coded Decimal Interchange Code). La codificación de los dos códigos anteriores coincide con la empleada en decimal empaquetado, lo que facilita la conversión.

Datos Lógicos

Es útil a veces considerar una unidad de n elementos o datos de 1 bit, cuando son tratados de esta manera, se consideran datos *lógicos*.

Tipos de Datos en el Pentium

Usa datos de 8 (byte), 16 (palabras), 32 (palabra doble), 64 (palabra cuádruple) bits de longitud. Para dar flexibilidad máxima en las estructuras de datos y una utilización eficiente de la memoria, las palabras no tiene por que estar alineadas con direcciones pares de memorias. El Pentium emplea el estilo “little-endian”, donde el byte menos significativo es almacenado en la dirección más baja.

Número de Direcciones	*Representación Simbólica*
General	Posiciones de byte, palabra, palabra doble y palabra cuádruple con contenido binario arbitrario.
Entero	Valor binario con signo dentro de un byte, palabra o palabra doble, representado en complemento a dos.
Ordinal	Entero sin signo en un byte, palabra o palabra doble.
BCD desempaquetado	Digito BCD, con un dígito por byte.
BCD empaquetado	Dos dígitos BCD en un byte, valor de 0 a 99.
Puntero de proximidad	Dirección de 32 bits.
Campo de bits	Secuencia contigua de bits, cada bit se considera como unidad independiente. Puede comenzar en cualquier posición de cualquier byte y puede contener hasta $2^{31}-1$ bytes.
Punto flotante	

Formatos de datos numéricos para la unidad de punto flotante del Pentium

Formatos de datos	*Rango*	*Precisión*	*Byte más significativo Dirección más alta*
Entero de una palabra	10^4	16 bits	15 —- 0 Comp. a 2
Entero Corto	10^9	32 bits	31 ——— 0 Comp. a 2
Entero Largo	10^{18}	64 bits	63 —————- 0 Comp. a 2
BCD Empaquetado	10^{18}	18 dígitos	s(79) X(78-72) 18 dígitos (71-0)
Precisión Simple	$10^{\pm 38}$	24 bits	s(31) Exp Sesg. (30-23) Mantisa (22-0)
Doble Precisión	$10^{\pm 308}$	53 bits	s(63) Exp. Sesg. (62-52) Mantisa (51-0)
Precisión Ampliada	$10^{\pm 4932}$	64 bits	s(79) Exp. Sesg. (78-64) Mantisa (63-0)

Tipos de Datos en el PowerPc

Puede manejar tipos de datos de 8 (byte), 16 (media palabra), 32 (palabra) y 64 (palabra doble) bits de longitud. En general no es necesario el alineamiento. Puede utilizar el estilo “little endian” o “big endian”.

El procesador interpreta los tipos de datos dependiendo de la instrucción. El procesador de punto fijo reconoce los siguientes tipos de datos:

- *Byte sin signo* Usado en operaciones lógicas o enteros.
- *Media palabra sin signo*.
- *Media palabra con signo*.
- *Palabra sin signo*.
- *Palabra con signo*.

- *Palabra doble sin signo*.
- *Cadena de bits*. De 0 a 128 bytes de longitud.
- *Punto flotante de precisión simple y doble* de acuerdo al IEEE 754.

Tipos de operaciones

Tipos generales de operaciones.

- Transferencia de datos
- Aritméticas.
- Lógicas.
- De conversión.
- De E/S.
- De control del sistema.
- De control de flujo.

Operaciones Usuales de Repertorios de Instrucciones

Tipo	*Nombre de la operación*
Transferencia	Move (transferir)
de datos	Store (guardar)
	Exchange (intercambiar fuente-destino)
	Clear (Transfiere una palabra de ceros)
	Set (Transfiere una palabra de unos)
	Push (Transfiere palabra desde fuente a la pila)
	Pop (Extrae palabra de la pila a un destino)
Aritméticas	Add
	Substract
	Multiply
	Divide
	Absolute
	Negate
	Increment
	Decrement

Tipo	*Nombre de la operación*
Lógicas	AND, OR, NOT (complemento) y Exclusive-OR a nivel bit.
	Test (comprueba condición indicada)
	Compare (comparación lógica o aritmética)
	Set Control Variables (gestión de interrupciones, temporizador, etc)
	Shift
	Rotate
Control de flujo	Jump (bifurcación)
	Jump Conditional (salto condicional)
	Jump to subrutine
	Return
	Exeute (no modifica el PC)
	Skip (salta la instrucción siguiente)
	Halt (parar)
	Wait (detiene la ejecución, comprueba repetitivamente y reanuda con condición satisfecha)
	No operation

Operaciones Usuales de Repertorios de Instrucciones

Tipo	*Nombre de la operación*
Entrada/	Input
Salida	Output
	Start I/O
	Test I/O
Conversión	Translate (traducción de valores usando tabla de correspondencia)
	Convert (Convierte contenido de una palabra de un formato a otro)

Acciones de la CPU para varios tipos de operaciones

Transferencia de datos	Transfiere datos de una posición a otra. Si es a memoria Determina la dirección Realiza la transformación de dirección de memoria virtual a real. Comprueba el cache e inicia la lectura/escritura en/de memoria.
Aritmética	Puede implicar transferencia de datos antes y/o después. Realiza la función en la ALU. Actualiza códigos e indicadores de condición
Lógica	Igual que una aritmética.
Conversión	Similar a la aritmética y la lógica, puede implicar a lógica especial para realizar la conversión.
Control de flujo	Actualiza el PC. En caso de llamadas y retornos de subrutinas, gestiona la trasferencia y enlace de parámetros.
E/S	Manda una orden a un módulo de E/S. En el caso de E/S asignada en memoria, determina la dirección de memoria correspondiente.

Control de Flujo

Por lo general, las instrucciones se ejecutan de posiciones consecutivas de memoria implícitamente, por lo que el CPU sólo incrementa el PC. Existen instrucciones que cambian la secuencia de ejecución, por lo que se actualiza el PC. Las razones por las que se necesitan las operaciones de control de flujo o de transferencia del control son:

1. Ejecutar instrucciones más de una vez.
2. Hacer algo cuando se cumple una condición, y otra cosa diferente si se cumple otra condición.
3. Es más fácil escribir un programa en módulos o fragmentos, en vez de un sólo programa largo.

Instrucciones de bifurcación Tienen como uno de sus operandos la dirección de la siguiente instrucción a ejecutar, las más frecuentes son las de *salto condicional*. Para comprobar la condición las máquinas proporcionan un código de condición de 1 o varios bits. El salto puede ser hacia adelante o hacia atrás.

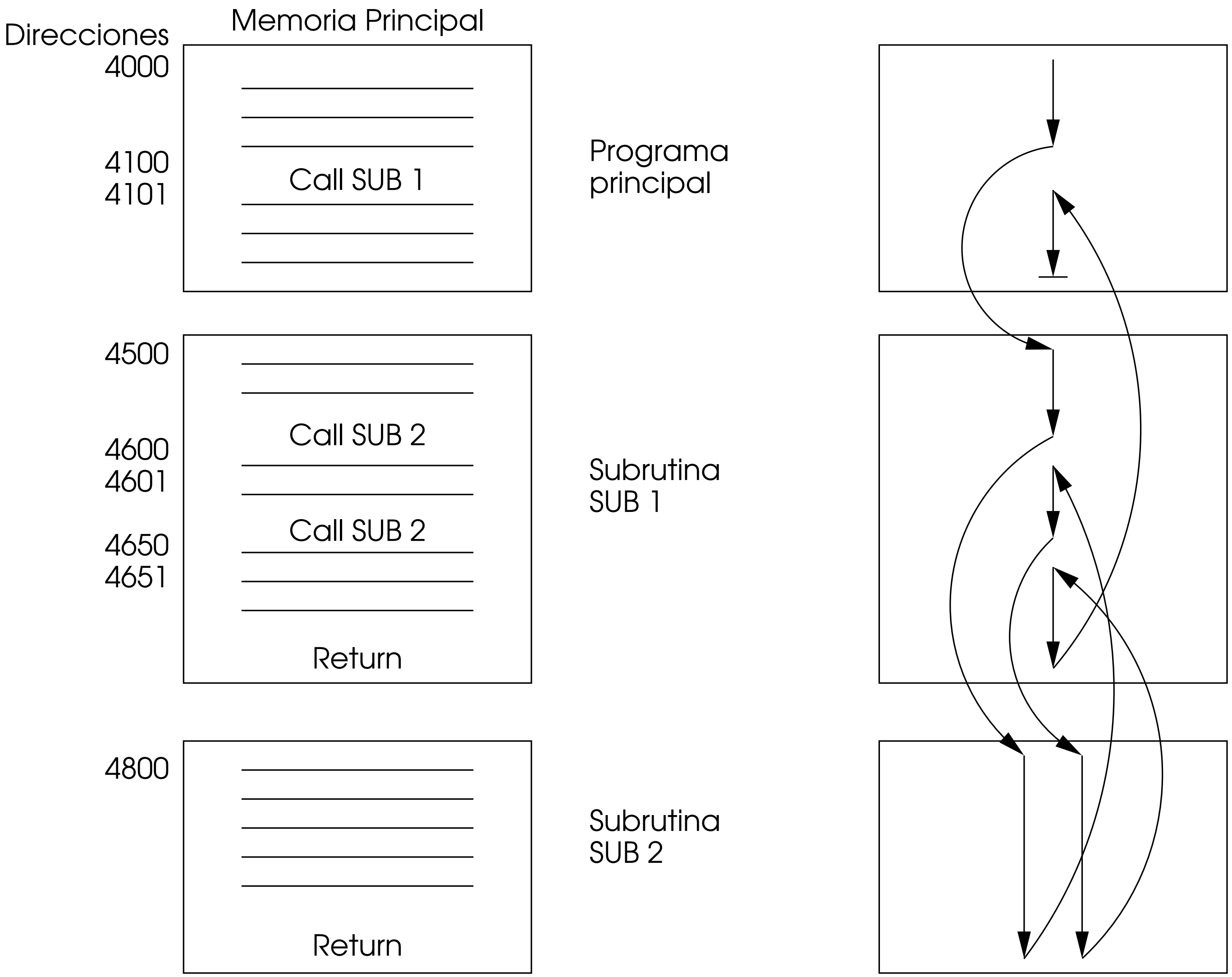
Instrucciones de salto implícito Incluye una dirección de manera implícita. La dirección es igual a la siguiente instrucción más la longitud de una instrucción, es decir, se salta una instrucción.

Instrucciones de llamada a subrutina Ordenan al CPU que pase a ejecutar una subrutina y después retornar al punto en que tuvo lugar la llamada. Su uso se justifica por la economía (uso eficiente del espacio de memoria) y la modularidad (facilidad para programar).

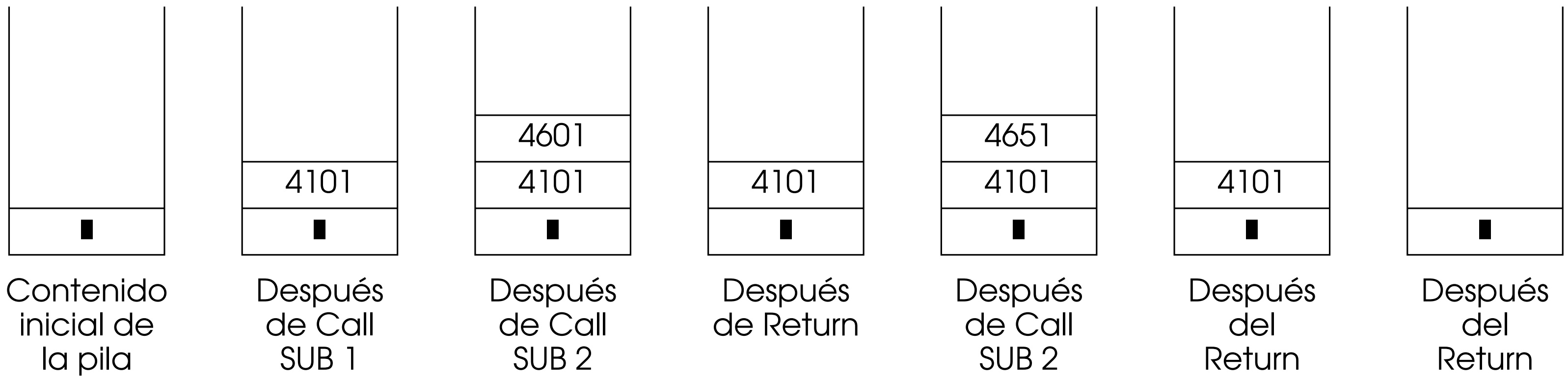
Además de aportar la dirección de retorno, es a menudo necesario transferir ciertos parámetros. Se pueden transferir usando registros y otra forma es almacenarlos en memoria. Las dos formas impiden el uso de subrutinas reentrantes.

Otra alternativa más flexible es la transferencia de parámetros a la pila, así cada vez que el procesador ejecuta una llamada, se introduce la dirección de retorno y los parámetros.

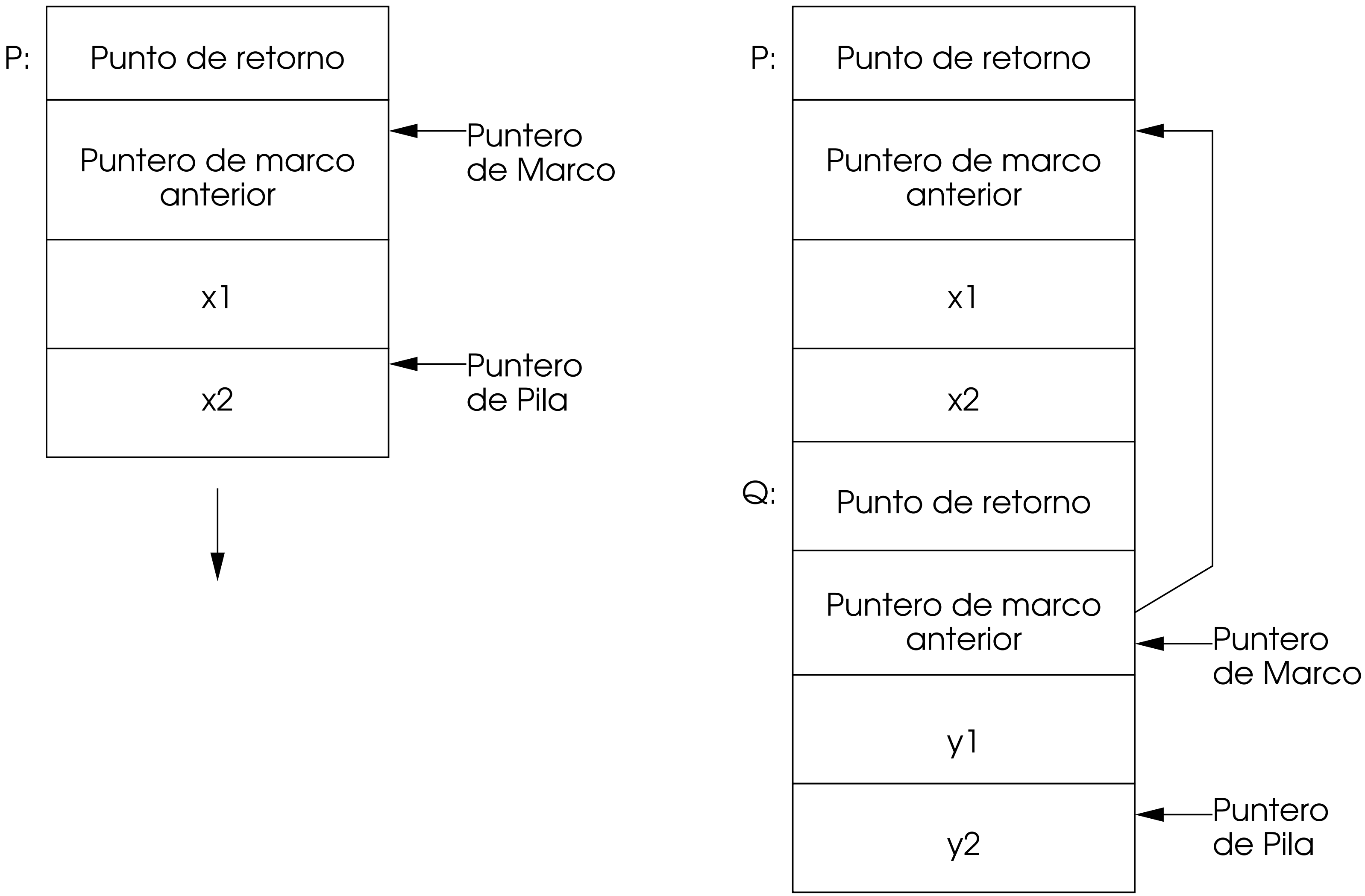
Subrutinas Anidadas y Secuencia de Ejecución



Uso de una pila para implementar el anidamiento



Crecimiento del marco de pila usando los procedimientos P y Q



Lenguaje Ensamblador

Un CPU interpreta y ejecuta instrucciones máquina, que son números binarios, por lo que un programador introduciría su programa como datos binarios.

Dirección	Contenido			
101	0010	0010	0000	0001
102	0001	0010	0000	0010
103	0001	0010	0000	0011
104	0011	0010	0000	0100
201	0000	0000	0000	0010
202	0000	0000	0000	0011
203	0000	0000	0000	0100
204	0000	0000	0000	0000

Programa en binario

Dirección		
101	LDA	201
102	ADD	202
103	ADD	203
104	STA	204
201	DAT	2
202	DAT	3
203	DAT	4
204	DAT	0

Programa Simbólico

Dirección	Contenido
101	2201
102	1202
103	1203
104	3204
201	0002
202	0003
203	0003
204	0000

Programa en Hexadecimal

Etiqueta	Operación	Operando
FORMUL	LDA	I
	ADD	J
	ADD	K
	STA	N
I	DATA	2
J	DATA	3
K	DATA	4
N	DATA	0

Programa en ensamblador

Cálculo de la fórmula $N = I + J + K$