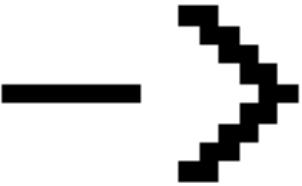


Organización del computador

ISA - Modos de direccionamiento

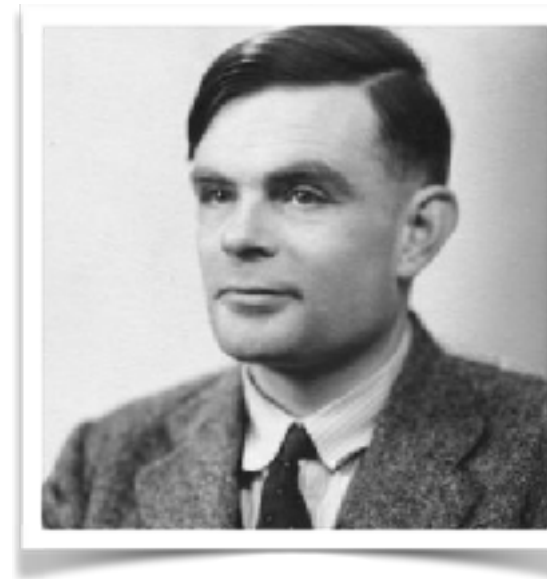
Jerarquía de máquina



Nivel 6	Usuario	Programa ejecutables
Nivel 5	Lenguaje de alto nivel	C++, Java, Python, etc.
Nivel 4	Lenguaje ensamblador	Assembly code
Nivel 3	Software del sistema	Sistema operativo, bibliotecas, etc.
Nivel 2	Lenguaje de máquina	Instruction Set Architecture (ISA)
Nivel 1	Unidad de control	Microcódigo / hardware
Nivel 0	Lógica digital	Circuitos, compuertas, memorias

- ➤ Cada nivel funciona como una máquina abstracta que oculta la capa anterior
- ➤ Cada nivel es capaz de resolver determinado tipo de problemas a partir de comprender un tipo de instrucciones específico
- ➤ La capa inferior es utilizada como servicio

Von Newman / Turing



- * Los programas y los datos se almacenan en la misma memoria sobre la que se puede leer y escribir
- * La operación de la máquina depende del estado de la memoria
- * **El contenido de la memoria es accedido a partir de su posición**
- * La ejecución es secuencial (a menos que se indique lo contrario)

Modos de direccionamiento

OpCode	Op1	Op2	Op3
--------	-----	-----	-----

- ➤ Los operandos de una instrucción pueden tener naturaleza variada:
 - ➤ Constantes,
 - ➤ Registros,
 - ➤ Referencias a la memoria: variables, etiquetas, nombres de subrutinas, etc.
- ➤ En todos los casos, salvo que se trate constantes o registros cuyo valor es usado como dato, los operandos son referencias a datos en la memoria

Modos de direccionamiento

General purpose registers (Orga1)

- ➤ Inmediato
- ➤ Directo (o absoluto)
- ➤ Indirecto
- ➤ Registro
- ➤ Indirecto con registro
- ➤ Desplazamiento (o indexado)

Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[c16]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

Modos de direccionamiento

Inmediato

- ➤ El operando es un valor y por lo tanto no requiere accesos adicionales a la memoria
- ➤ **Ejemplos:**
 - ➤ MOV R1, 230

0001	100001	000000	(230) ₂	
------	--------	--------	--------------------	--

Modos de direccionamiento

Directo (o absoluto)

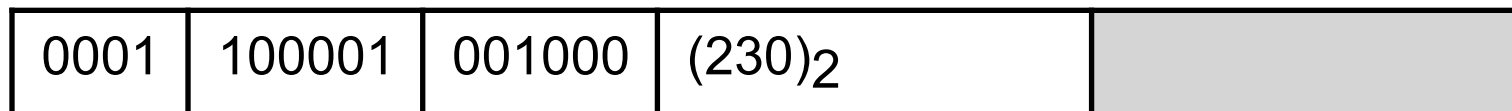
- ➤ El valor se encuentra en la dirección de memoria que figura como operando en la instrucción
- ➤ Es el tipo de acceso que se utiliza cuando se usan variables
- ➤ **Ejemplo:**
 - ➤ MOV R1, Var

0001	100001	001000	(230) ₂	
------	--------	--------	--------------------	--

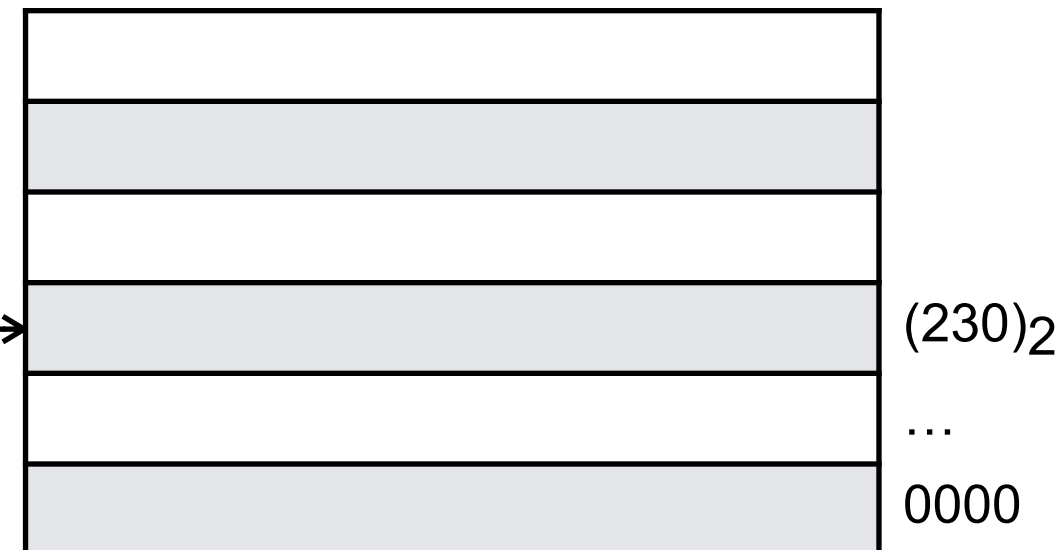
Modos de direccionamiento

Directo (o absoluto)

Instrucción



Memoria



Modos de direccionamiento

Indirecto

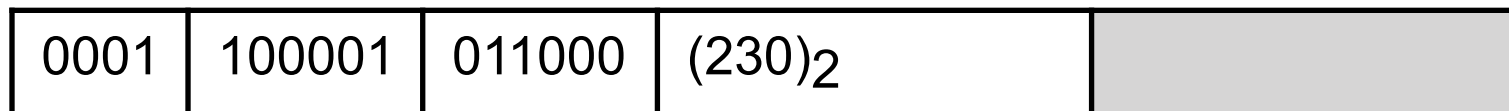
- ➤ El valor se encuentra en la dirección de memoria que se encuentra en la dirección de memoria que figura como operando en la instrucción
- ➤ Es el tipo de acceso que se utiliza cuando se usan punteros
- ➤ Requiere más de un acceso a la memoria en la etapa de búsqueda de operandos del ciclo de instrucción
- ➤ **Ejemplo:**
 - ➤ MOV R1, [Var]

0001	100001	011000	(230) ₂	
------	--------	--------	--------------------	--

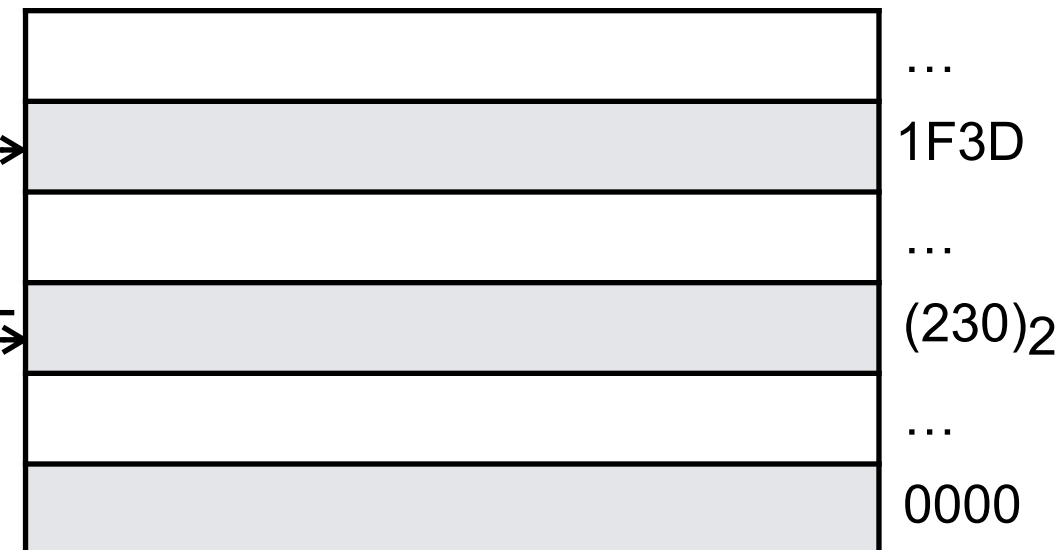
Modos de direccionamiento

Directo (o absoluto)

Instrucción



Memoria



Modos de direccionamiento

Registro

- ➤ El valor se encuentra en el registro
- ➤ Es el tipo de acceso que se utiliza cuando se usan variables
- ➤ **Ejemplo:**
 - ➤ MOV R1, R5

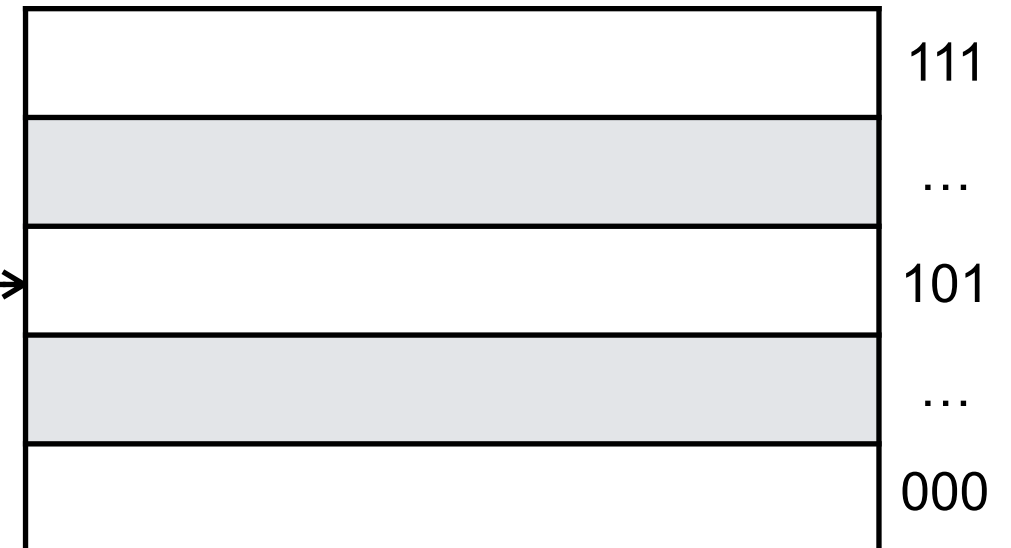
0001	100001	100101		
------	--------	--------	--	--

Modos de direccionamiento

Registro

Instrucción

Registros



Modos de direccionamiento

Indirecto con registro

- ➤ El valor se encuentra en la dirección de memoria que está en un registro
- ➤ Es el tipo de acceso que se utiliza cuando se usan arreglos
- ➤ **Ejemplo:**
 - ➤ `MOV R1, [R5]`

0001	100001	110101		
------	--------	--------	--	--

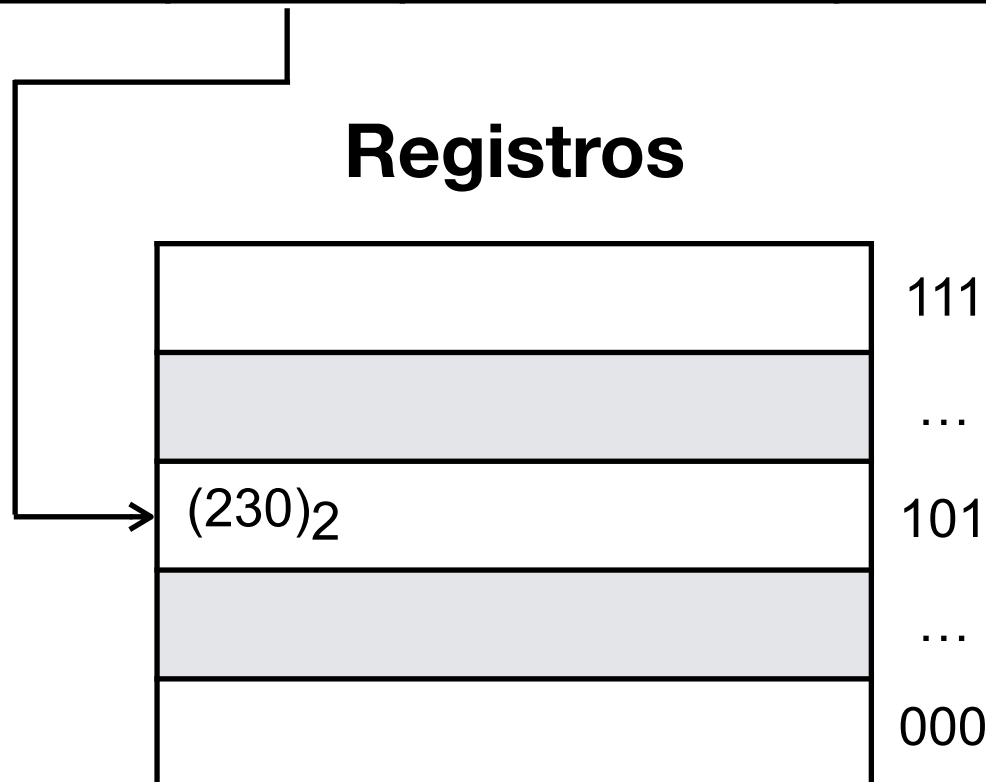
Modos de direccionamiento

Indirecto con registro

Instrucción



Registros



111

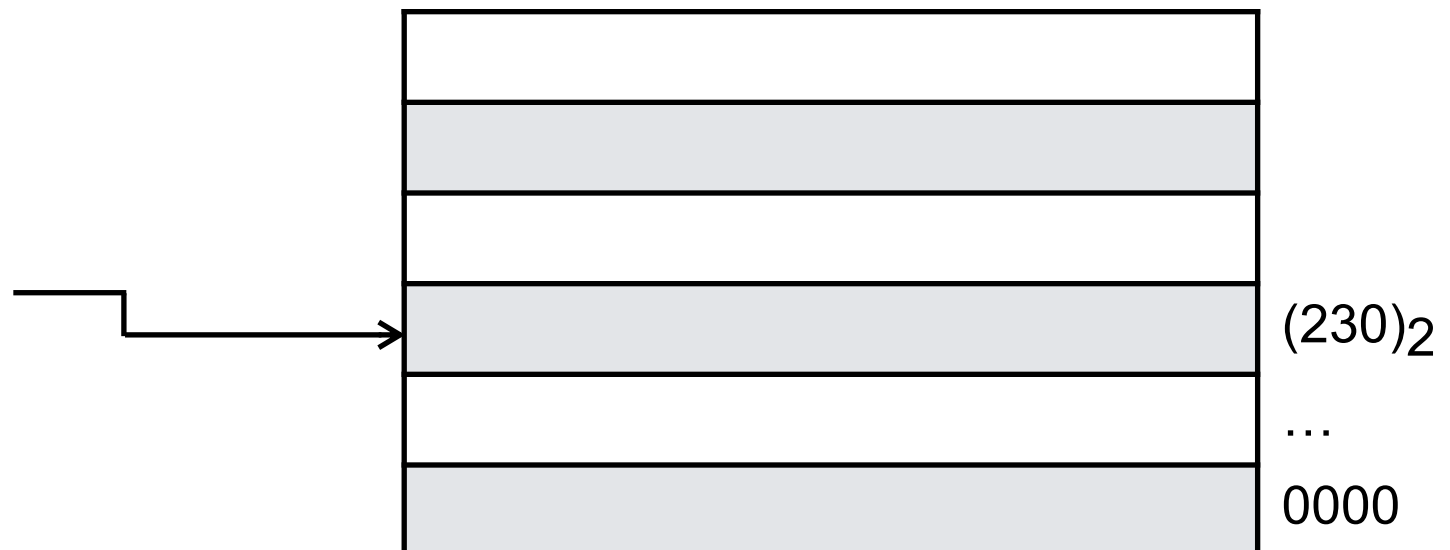
...

101

...

000

Memoria



(230)₂

...

0000

Modos de direccionamiento

Desplazamiento (o indexado)

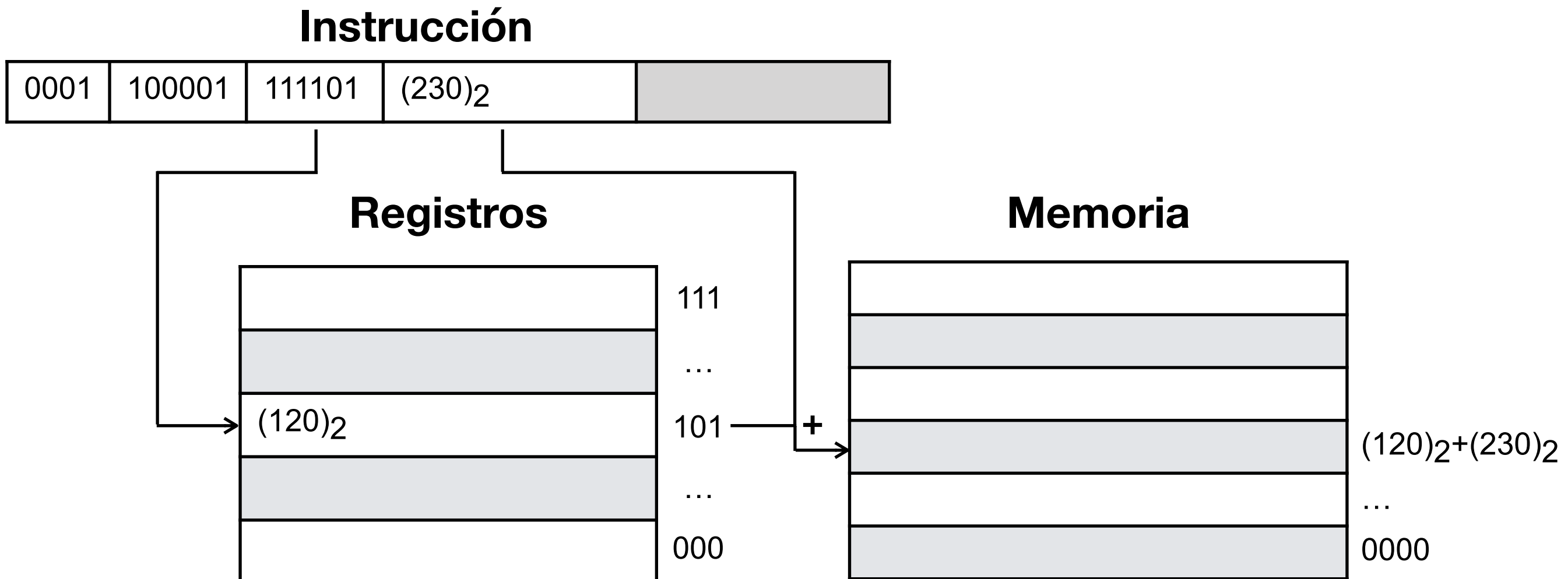
- ➤ El valor se encuentra en la dirección de memoria que está en un registro más la constante que acompaña la instrucción como operando
- ➤ Es el tipo de acceso que se utiliza cuando se accede a campos de una estructura
- ➤ **Ejemplo:**

— ➤ MOV R1, [R5 + (230)₂]

0001	100001	111101	(230) ₂	
------	--------	--------	--------------------	--

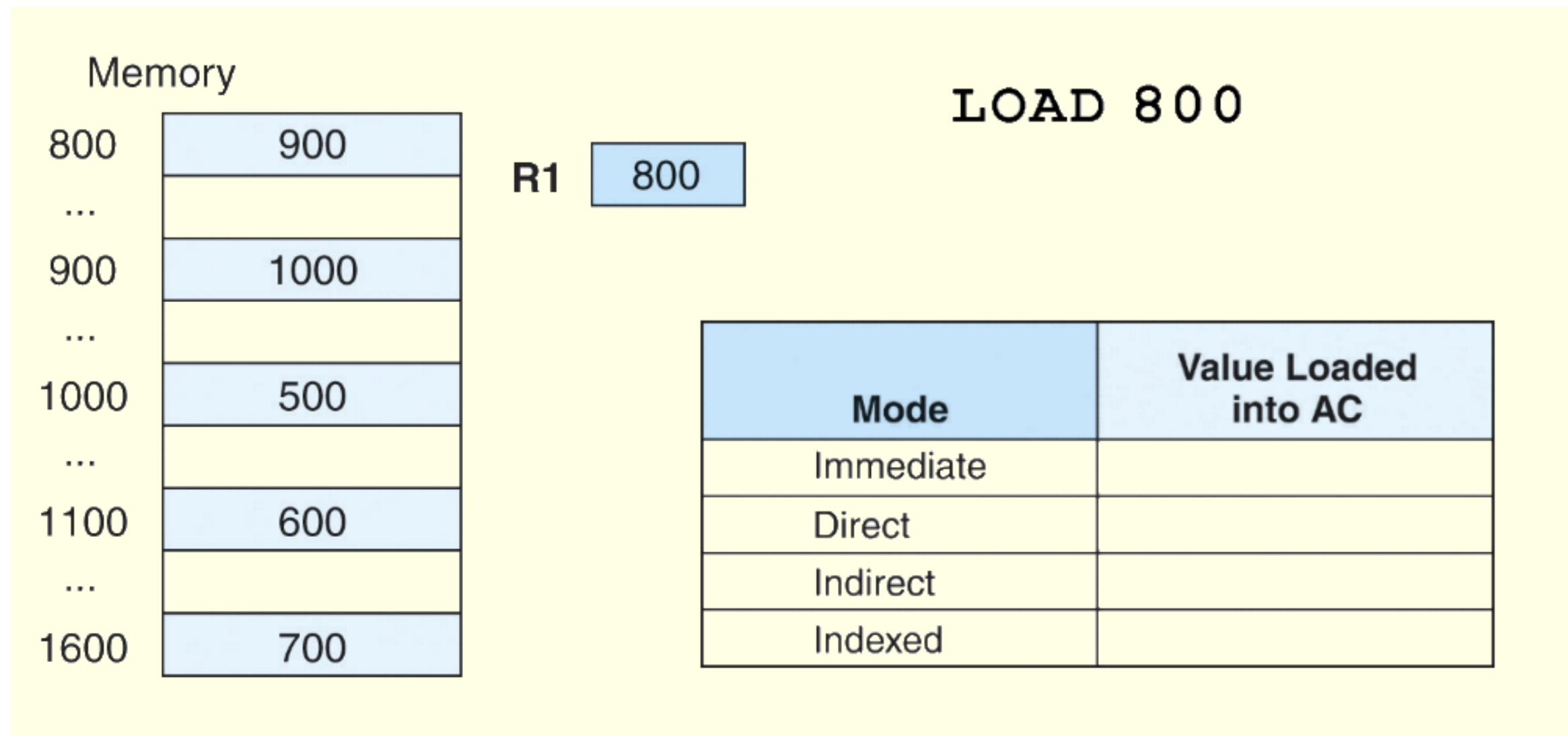
Modos de direccionamiento

Desplazamiento (o indexado)



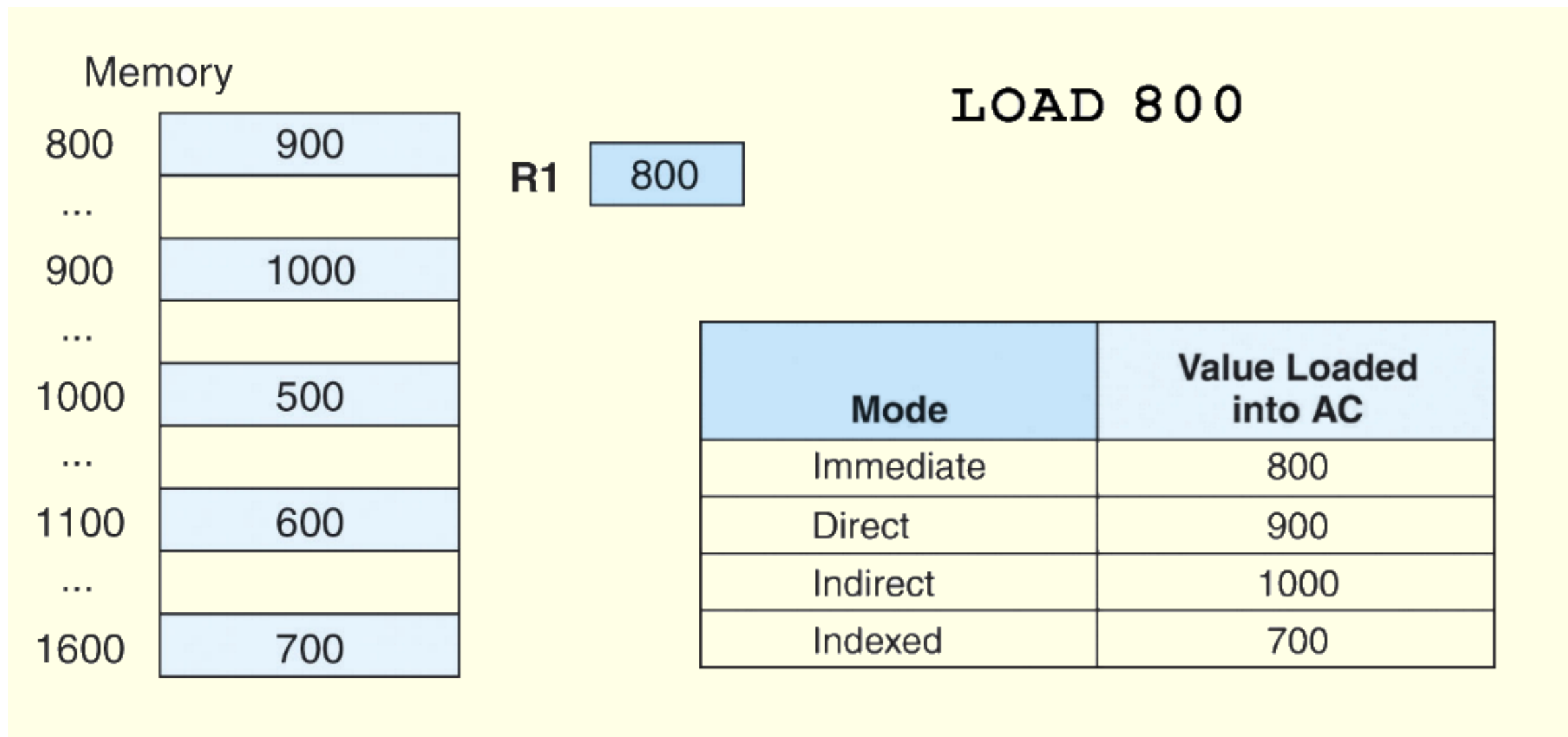
Modos de direccionamiento

Ejemplos



Modos de direccionamiento

Ejemplos



Modos de direccionamiento

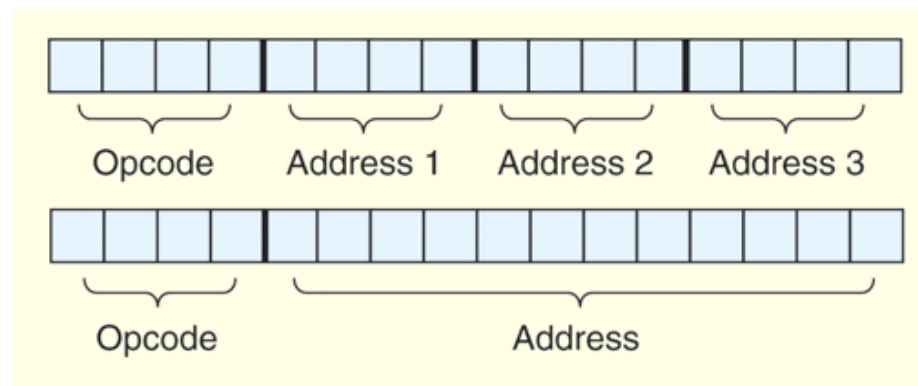
Ortogonalidad

- ➤ Refiere a la posibilidad de utilizar cualquier modo de direccionamiento con cualquier operación
- ➤ Resulta en un número muy grande de instrucciones: n instrucciones y m modos de direccionamiento, que pueden combinarse entre sí, arroja $n \times 2^m$ instrucciones diferentes
- ➤ Es una cualidad elegante para el código pero muy costosa para la implementación de la unidad de control
- ➤ Puede que muchas de estas instrucciones sean realmente utilizadas como para justificar su costo en complejidad de la unidad de control

Modos de direccionamiento

Ortogonalidad

- ➤ No todas las operaciones requieren de la misma cantidad de operandos:

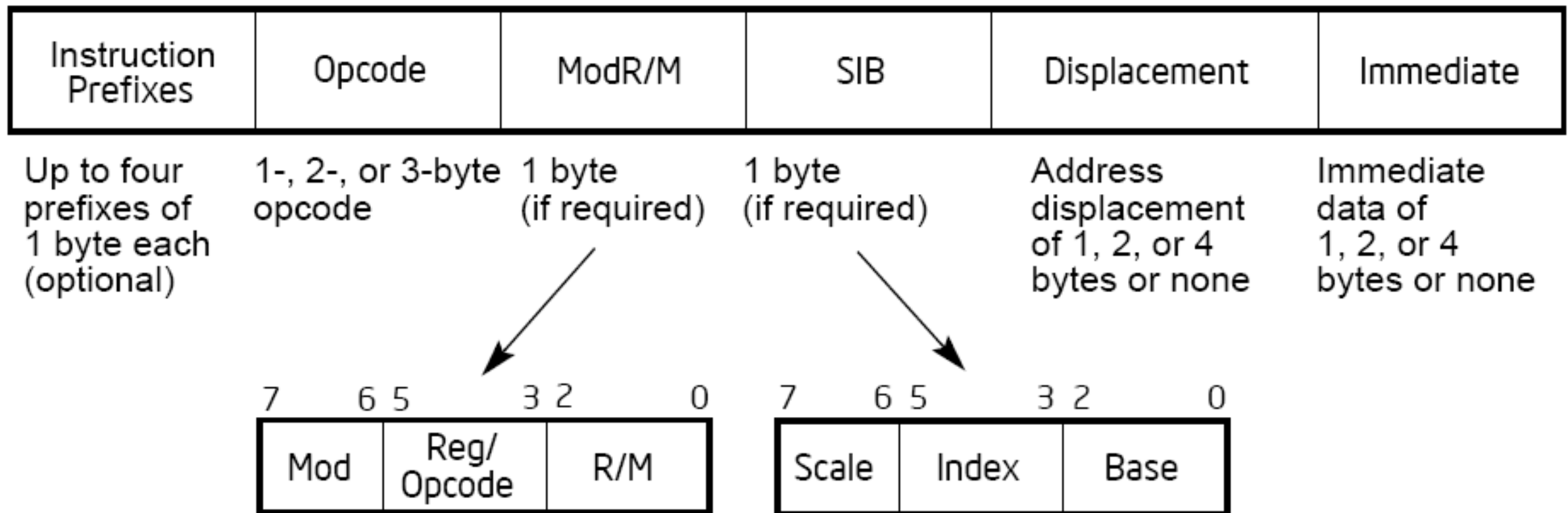


- ➤ Se utiliza códigos de operación variables para expandir la cantidad posible de operaciones

0000	R1	R2	R3	}	15 3-address codes
1110	R1	R2	R3		
1111	0000	R1	R2	}	14 2-address codes
1111	1101	R1	R2		
1111	1110	0000	R1	}	31 1-address codes
1111	1111	1110	R1		
1111	1111	1111	0000	}	16 0-address codes
1111	1111	1111	1111		

Modos de direccionamiento

Pentium (Formato de instrucción)



Modos de direccionamiento

Pentium (Modo de direccionamiento)

- ➤ Se dirección usando un Segmento sobre el que luego se calcula una dirección lineal dentro de dicho segmento
- ➤ Modos de direccionamiento lineal:
 - ➤ Inmediato
 - ➤ Implícito
 - ➤ Register operand
 - ➤ Displacement
 - ➤ Base
 - ➤ Base with displacement
 - ➤ Scaled index with displacement
 - ➤ Base with index and displacement
 - ➤ Base scaled index with displacement
 - ➤ Relative

Modos de direccionamiento

Pentium (Modo de direccionamiento)

