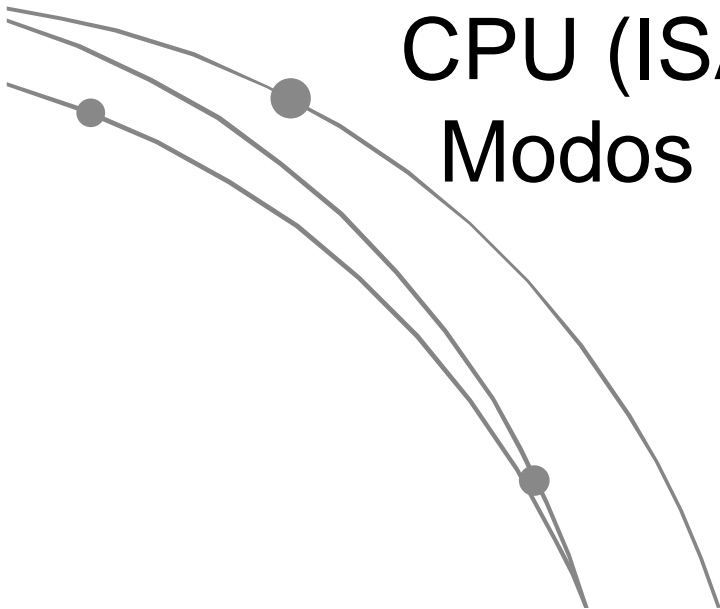


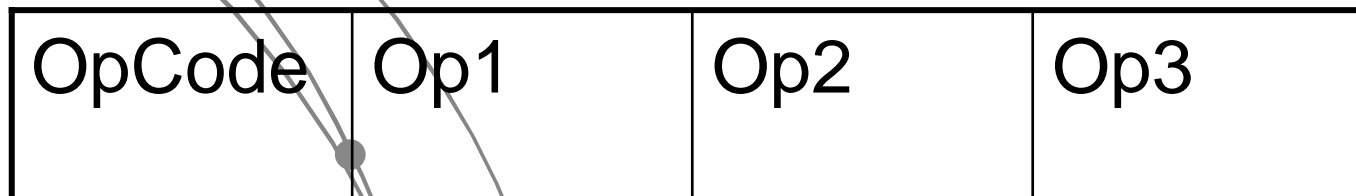
Organización del Computador

CPU (ISA)– Diseño de un ISA
Modos de Direcccionamiento




Modos de Direcccionamiento

- Instrucción: OpCode + Operandos
- Que tipos de cosas pueden ser los operandos?
 - Constantes
 - Referencia a Variables
 - Referencia a Arrays
 - Referencias a subrutinas
 - Estructuras de datos (Listas, Colas)



Modos de Direccionamiento

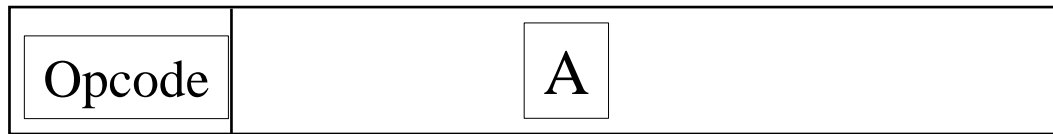
- Inmediato
 - Directo (o absoluto)
 - Indirecto
 - Registro
 - Indirecto con registro
 - Desplazamiento (Indexado)
- 
- Three decorative curved lines in a light gray color sweep from the left side of the slide towards the bottom right. Three small gray dots are placed along these lines: one on the top curve, one on the middle curve, and one on the bottom curve.

Inmediato



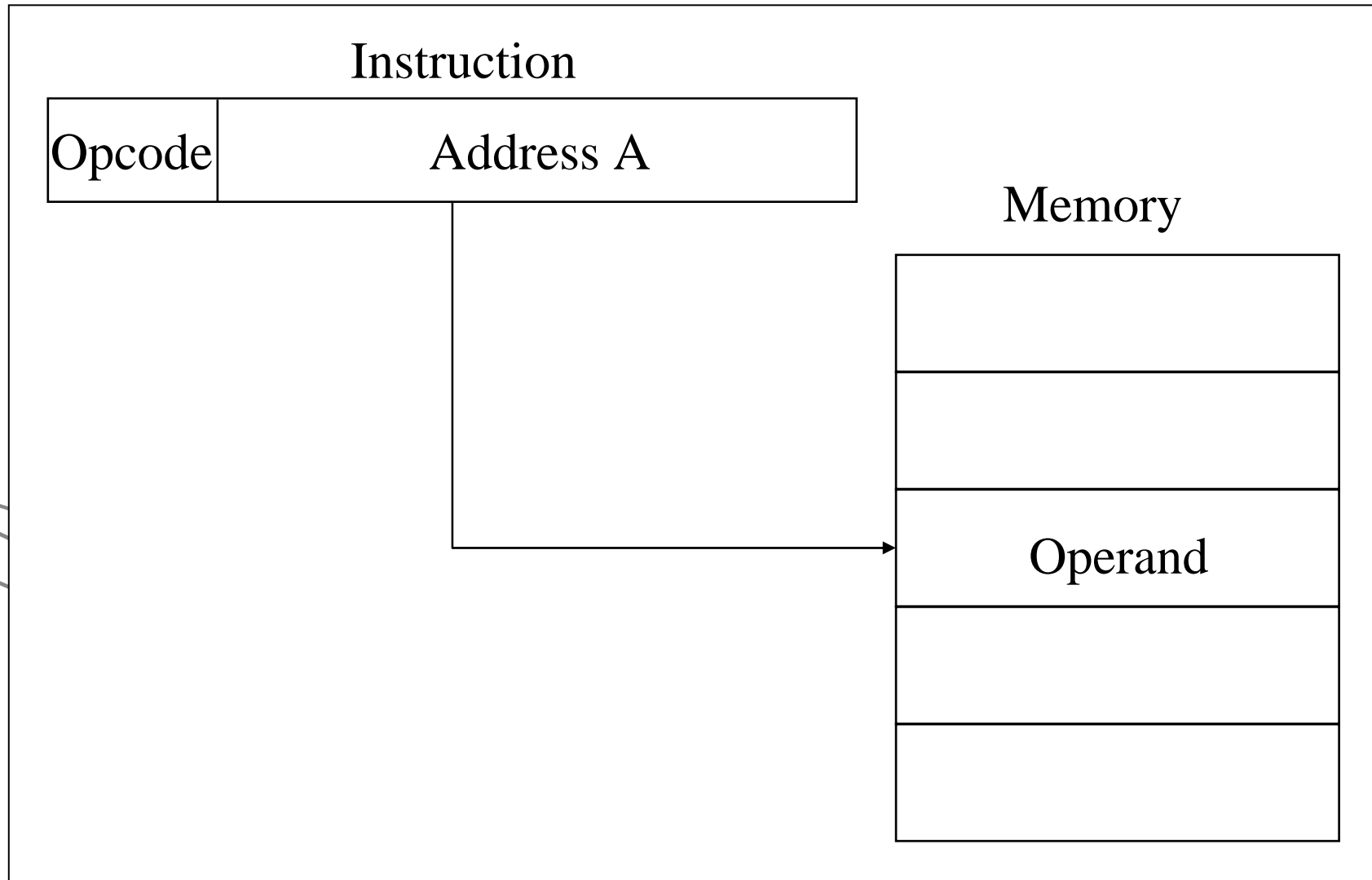
- El operando es parte de la instrucción (N)
- Ej: ADD 5
 - $AC = AC + 5$
 - 5 es un operando
- Ej2: Jump 110
- No hay referencia adicional a memoria
- Rápido
- **Rango limitado**

Directo



- El operando está en la dirección referenciada por A
- Operando = [A]
- Ej: ADD [941] ($A = A + [941]$)
- Ideal para acceso a variables
- Hay sólo un acceso a la memoria
- Direccionamiento limitado a tamaño del operando

Directo

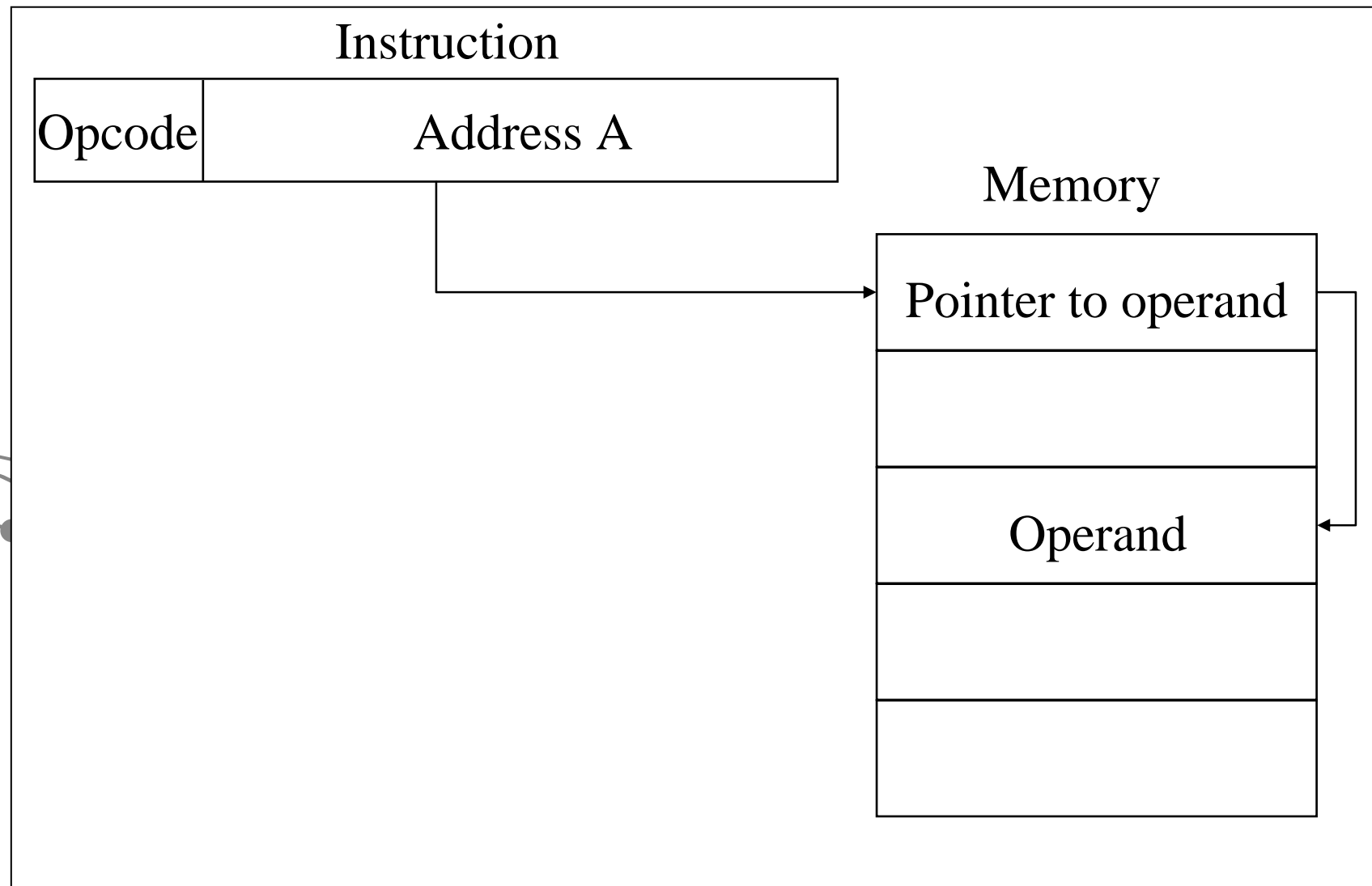


Indirecto



- A es un Puntero
- Operando = $[[A]]$
- Usos
 - Acceso a Arrays, Listas u otras estructuras
 - Aumenta el espacio de direccionamiento
- Existe acceso múltiple a la memoria para encontrar el operando

Indirecto



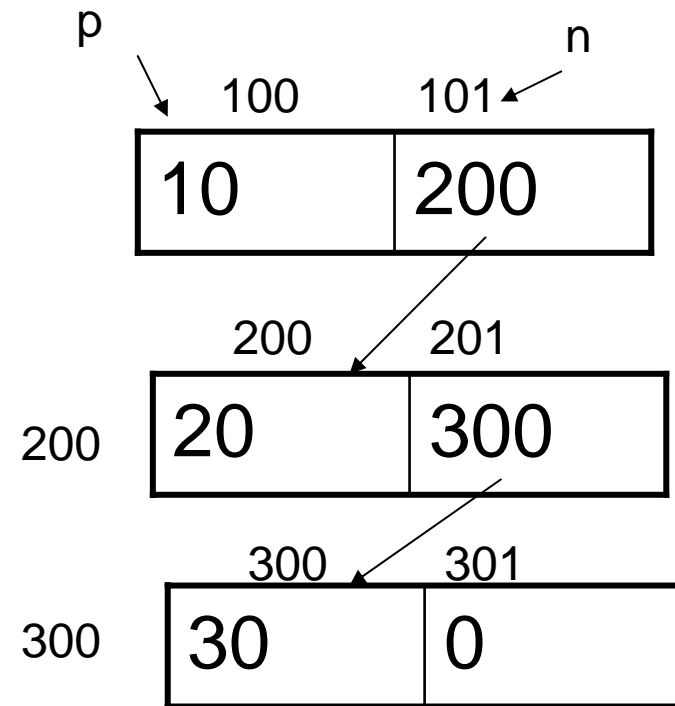
Ejemplo Lista indirecto

```
      Mov R1,0
Start: cmp [p],0
      je fin
      add R1,[p]
      mov [p],[[n]]
      mov [n],[p]
      add [n],1
      jmp Start
```

End:

p: DW 100

n: DW 101

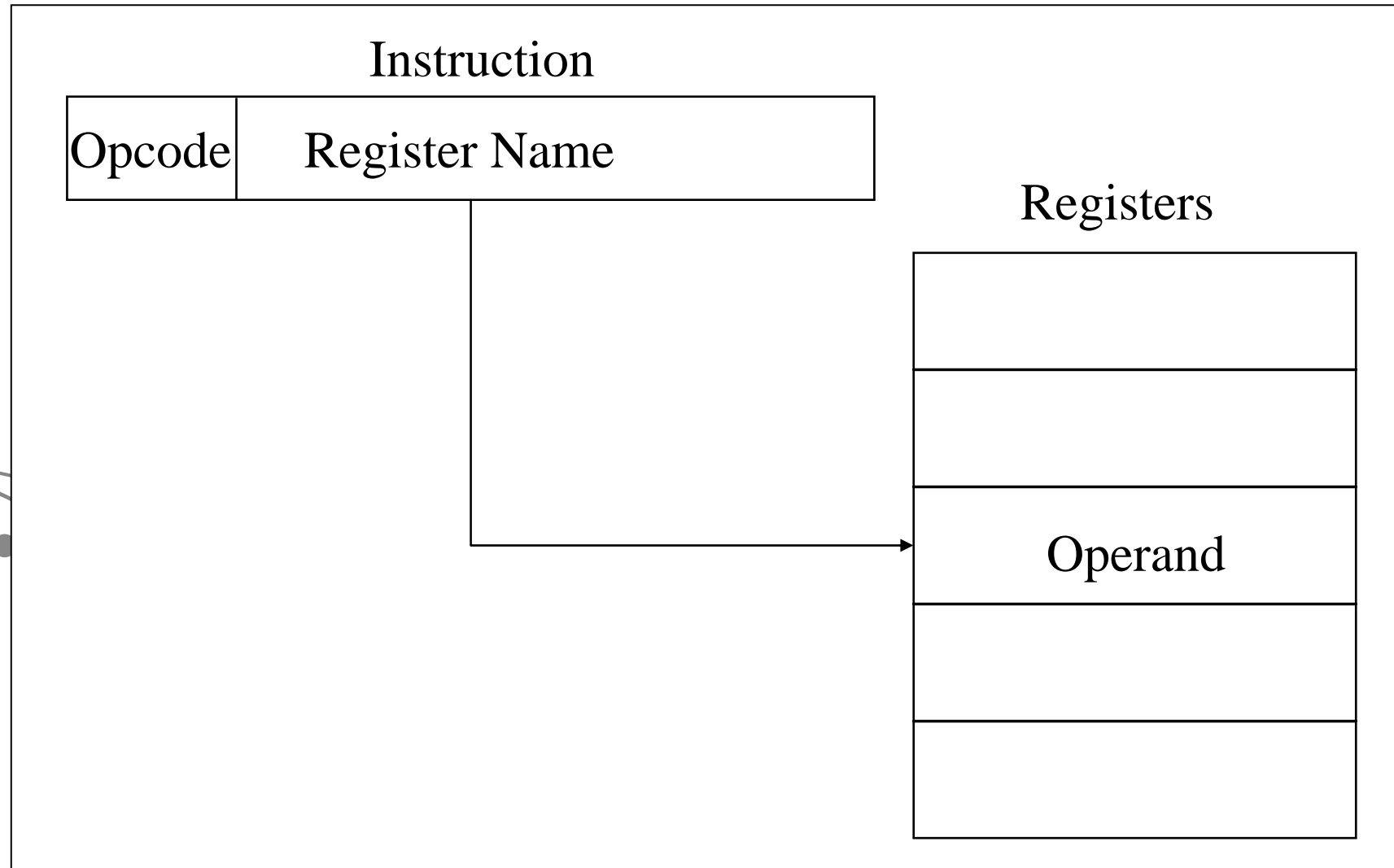


Registro



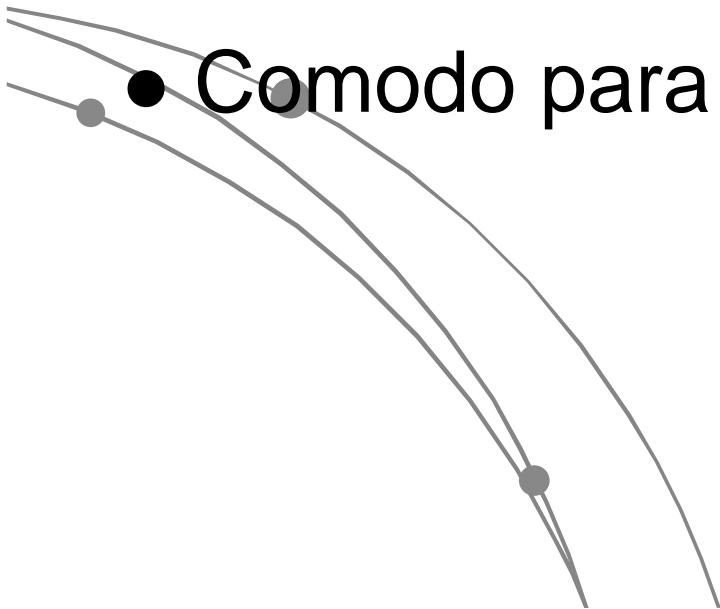
- El operando es un registro de la CPU
- Operando = Registro n
- Número limitado de registros
- Instrucción rápida
 - Ej: Mov R1,R2
- No acceso a memoria
- Instrucción corta
- Espacio de direcciones limitado

Registro

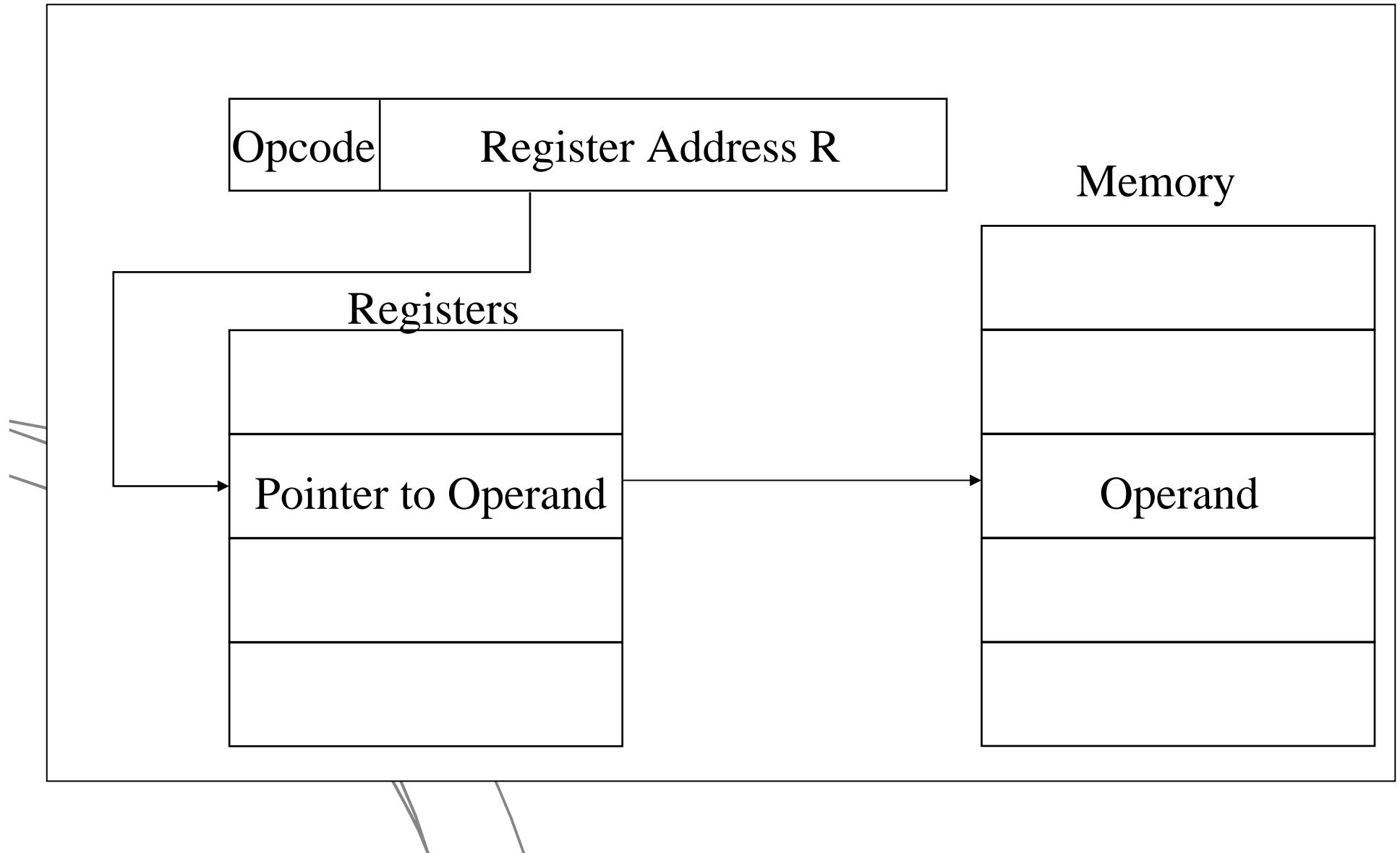


Registro Indirecto

- El operando está en la memoria direccionada por un registro.
- Operando = $[R_n]$
- Hay un acceso menos a memoria que en direccionamiento indirecto
- Comodo para acceder a arrays



Registro Indirecto

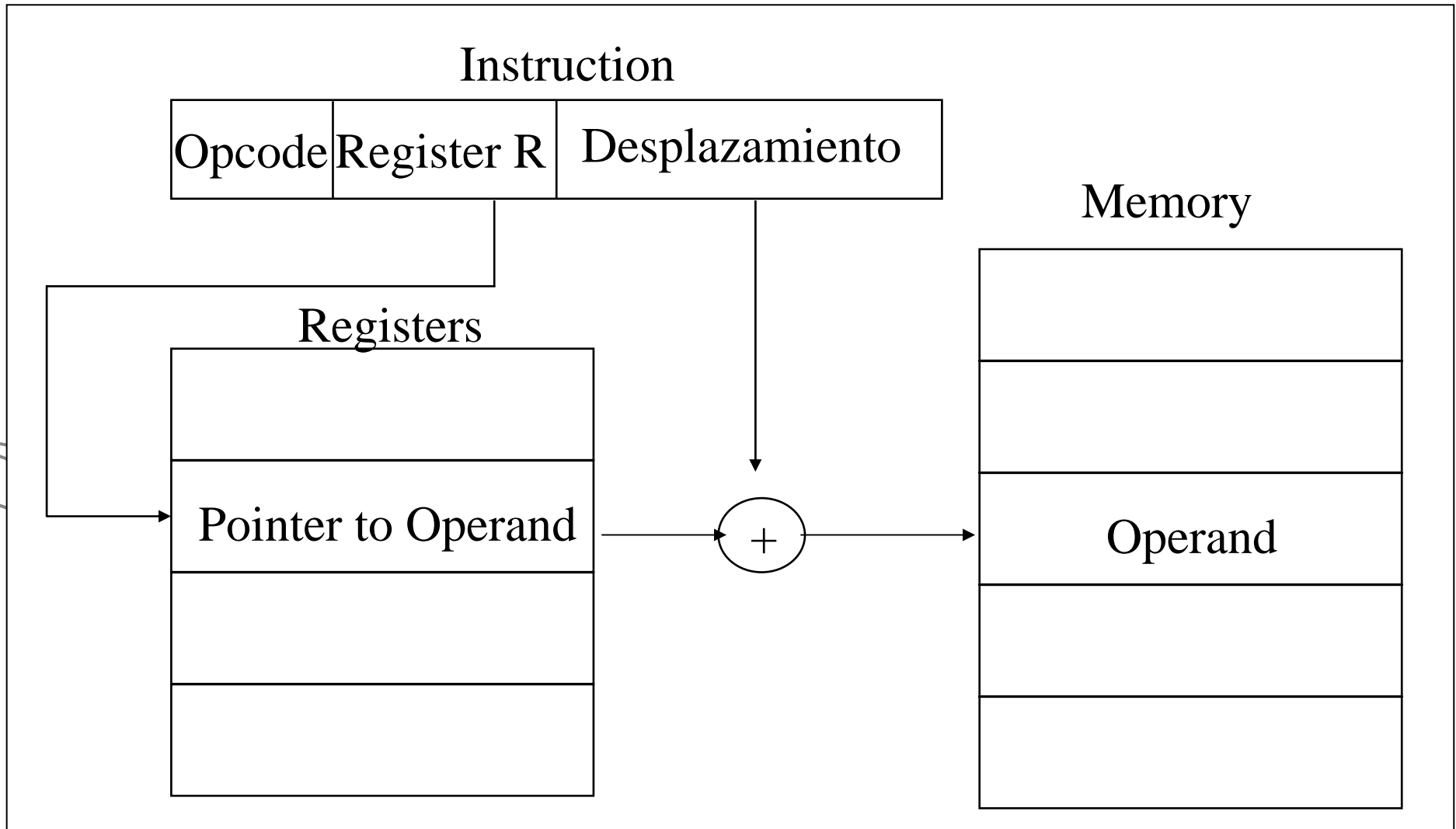


Desplazamiento

- El operando contiene una referencia a un registro y a un valor de desplazamiento
- Operando = $[R_{N1} + D]$
- Ideal para acceder a campos de registros
 - Moviendo D
- También para arrays de estructuras
 - R se mueve dentro del array
 - D selecciona el campo



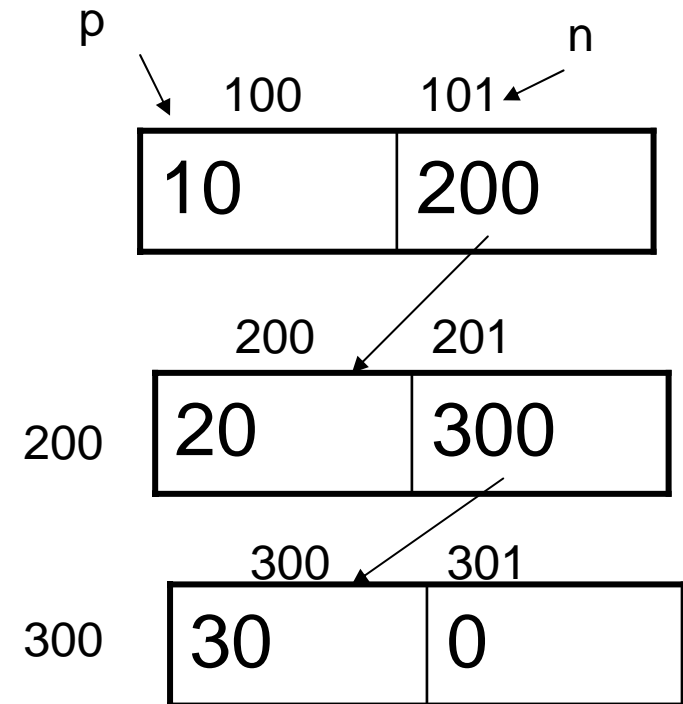
Desplazamiento



Ejemplo Lista sin desplazamiento

```

Mov R1,0
Mov R2,[p]
Mov R3,[n]
Start: cmp [r2],0
      je fin
      add R1,[R2]
      mov R2,[R3] // R2=[ [n] ]
      mov R3,R2
      add R3,1    // R3 = [n]
      jmp Start
End:
p:   DW 100
n:   DW 101
```

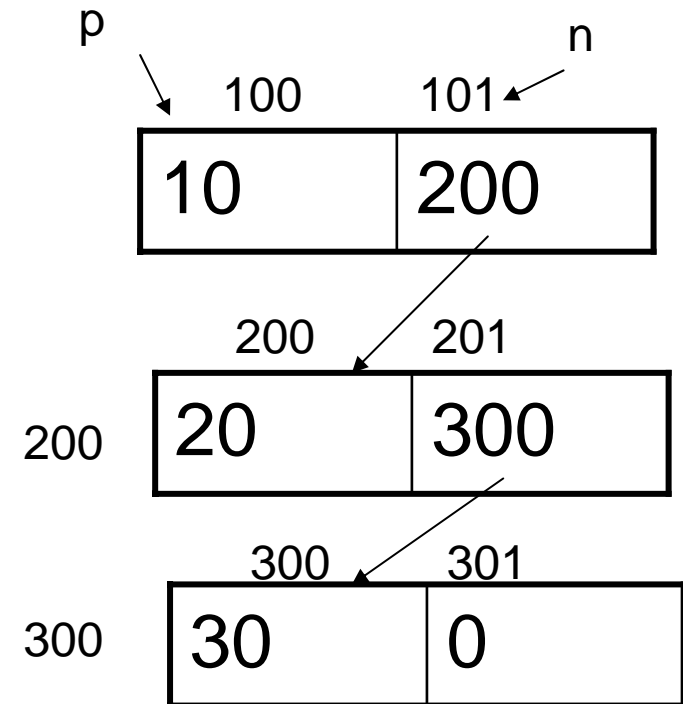


Ejemplo Lista con desplazamiento

```
      Mov R1,0
      Mov R2,[p]
Start: cmp [r2],0
      je fin
      add R1,[R2]
      mov R2,[R2+1]; R2=[[n]]
      jmp Start
```

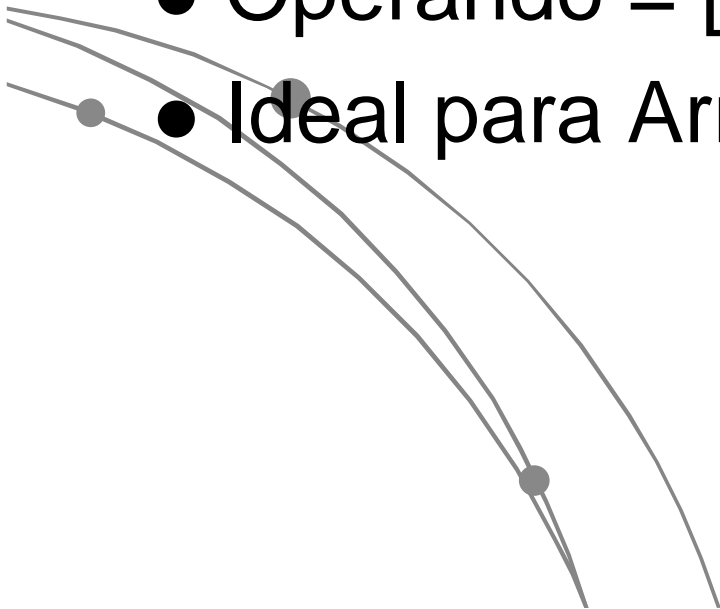
End:

p: DW 100



Indexado

- Similar al desplazamiento
- Un operando contiene una referencia a una dirección y a un registro que actúa como desplazamiento
- Operando = $[D+R_{N1}]$
- Ideal para Arrays



Modo de Direcccionamiento	
Inmediato	Mov R, #N $R \leftarrow N$
Directo	Mov R, [A] $R \leftarrow \text{mem}[A]$
Register	Mov R1,R2 $R1 \leftarrow R2$
Register Indirect	Mov R1,[R2] $R1 \leftarrow \text{mem}[R2]$
Displacement	Mov R1, [R2+D] $R1 \leftarrow \text{mem}[R2 + D]$
Base-Register Displacement	Mov R1, D (R_{base}) $R1 \leftarrow \text{mem}[R_{\text{base}} + D]$
Indexed	Mov R1, A[R] $R1 \leftarrow \text{mem}[A+R]$
Indexed Scaled	Mov R1, A[R*Scale] $R1 \leftarrow \text{mem}[A+R*\text{Scale}]$
PC Relative	Jump N $PC \leftarrow PC + N$

Ejemplo

- Completar el valor de AC según el modo de direccionamiento

Memory

800	900
...	
900	1000
...	
1000	500
...	
1100	600
...	
1600	700

R1 800

LOAD 800

Mode	Value Loaded into AC
Immediate	
Direct	
Indirect	
Indexed	

Ejemplo

Memory

800	900
...	
900	1000
...	
1000	500
...	
1100	600
...	
1600	700

R1

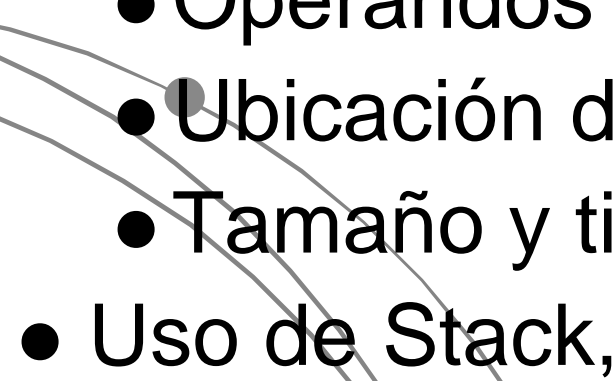
800

LOAD 800

Mode	Value Loaded into AC
Immediate	800
Direct	900
Indirect	1000
Indexed	700

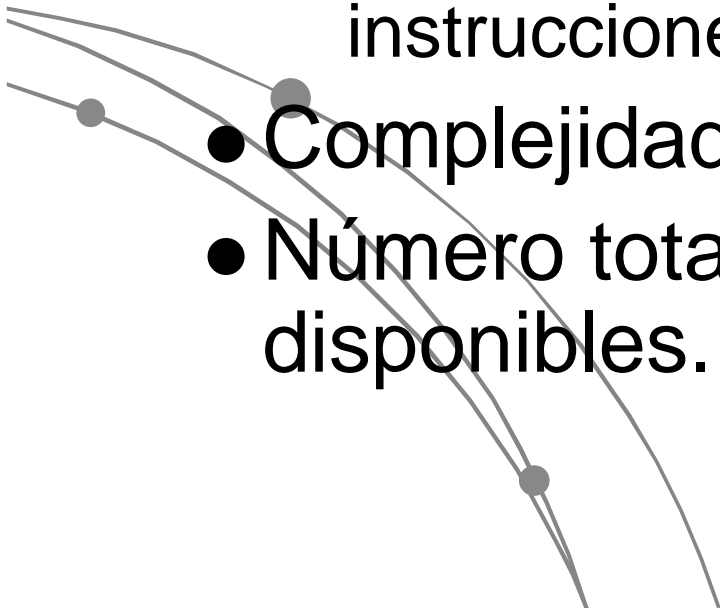
Diseñando el ISA

Temas a considerar:

- Tipos de operación
 - Número de bits por instrucción
 - Número de operandos por instrucción
 - Operandos implícitos y explícitos
 - Ubicación de los operandos
 - Tamaño y tipo de los operandos
 - Uso de Stack, Registros
- 

Diseñando el ISA

- Algunas métricas...
 - Memoria principal ocupada por el programa
 - Tamaño de la instrucción (en bits).
 - Code density: tratar que las instrucciones ocupen poco
 - Complejidad de la instrucción.
 - Número total de instrucciones disponibles.



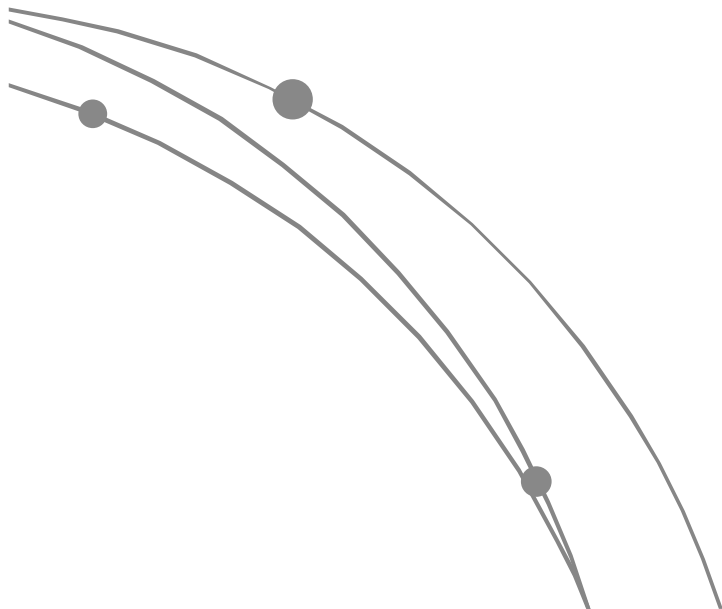
Criterios en diseños de ISA

Tamaño de la instrucción

- Corto, largo, variable?

Cuántos operandos?

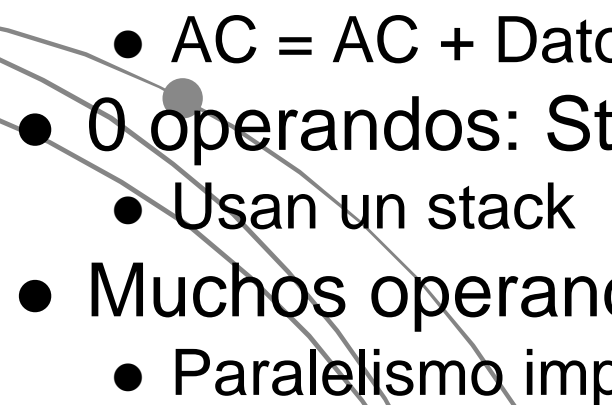
Cuántos registros?



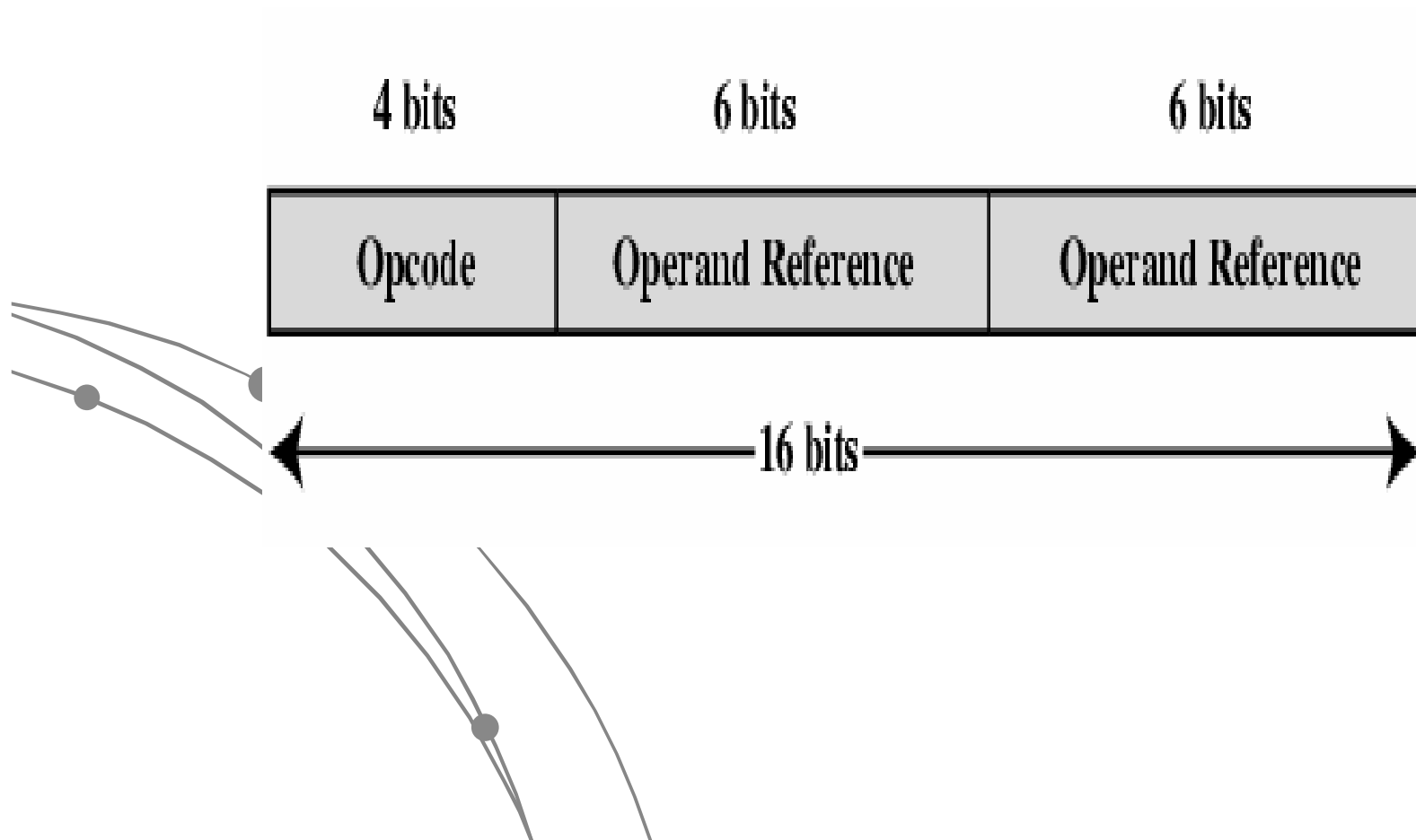
Memoria

- Direccionable por byte o por palabra (word)?
- Big/Little Endian
- Cuántos modos de direccionamiento?
 - Directo, indirecto, indexado...
 - Muchos
 - Pocos
 - Uno solo

Cuántos Operandos?

- 3 operandos: RISC y Mainframes
 - $A = B + C$
 - 2 operandos: Intel, Motorola
 - $A = A + B$
 - Uno suele ser un registro
 - 1 operando: Acumulador
 - $AC = AC + \text{Dato}$
 - 0 operandos: Stack Machines
 - Usan un stack
 - Muchos operandos (VLIW)
 - Paralelismo implícito en la instrucción
- 

Formato Posible

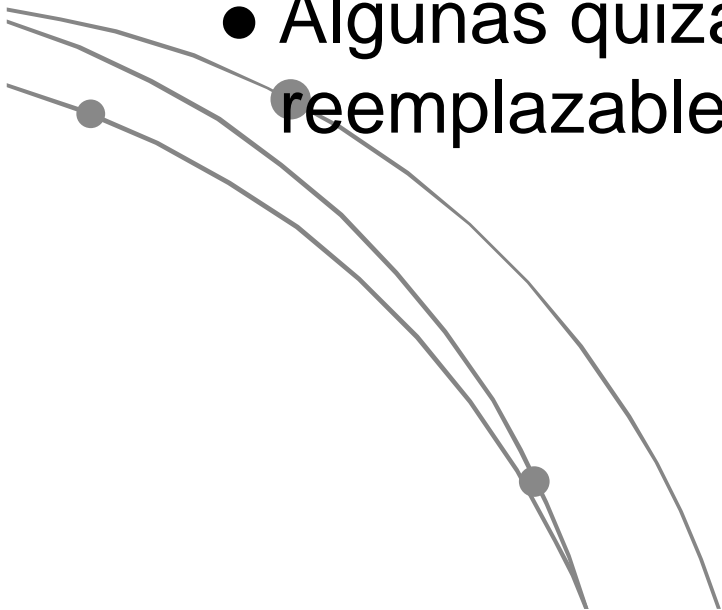


Algunas ISA

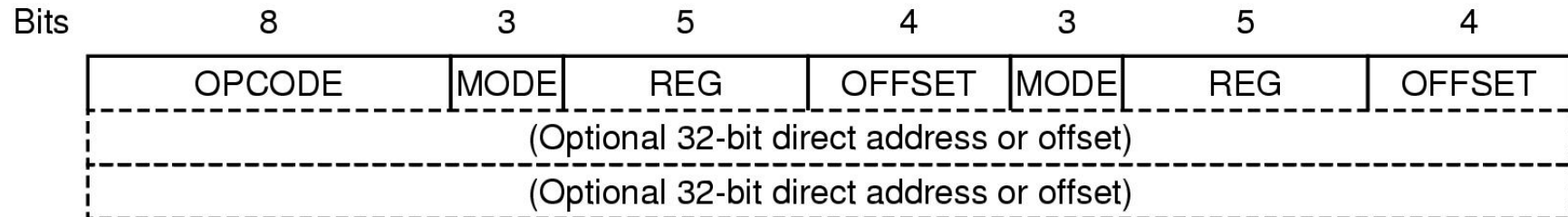
	CISC examples			RISC examples		Superscalars	
	IBM 370/168	VAX 11/780	Intel 80486	88000	R4000	RS/6000	80960
Year developed	1973	1978	1989	1988	1991	1990	1989
The number of instruction	208	303	235	51	94	184	62
Instruction size (bytes)	2 - 6	2 - 57	1 - 11	4	4	4	4, 8
Addressing modes	4	22	11	3	1	2	11
The number of GRPs	16	16	8	32	32	32	32 - 256
Control memory size (K bits)	420	480	246	-	-	-	-
Cache size (KB)	64	64	8	16	128	32 - 64	0.5

Ortogonalidad

- Cualquier instrucción puede ser usada con cualquier modo de direccionamiento
- Es una cualidad “elegante”, pero costosa
 - Implica tener muchas instrucciones
 - Algunas quizás poco usadas o fácilmente reemplazables



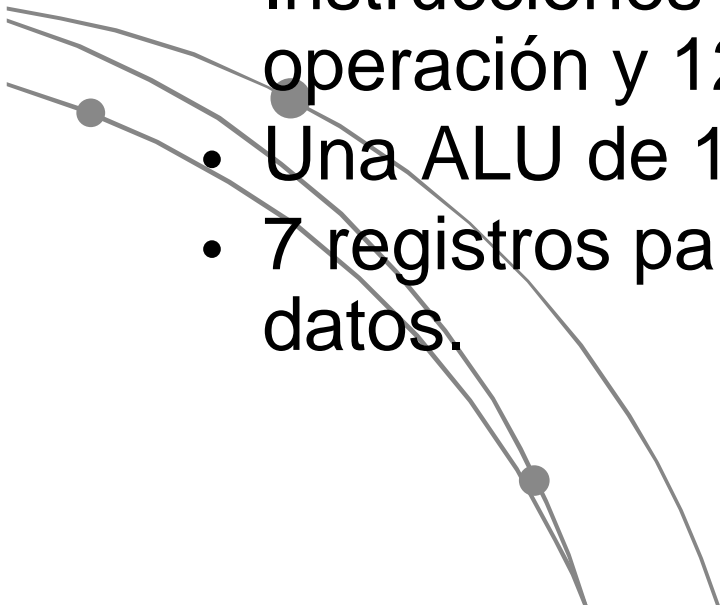
Ejemplo



- Instrucción 32 bits
- 256 instrucciones posibles
- 2 operandos (Op1 = Op1 op Op2)
- 12 bits por operando
 - MODE = 8 Modos de Direcccionamiento
 - Reg = 32 Registros
 - OffSet = Desplazamiento o Escala (4 bits)
- Problema: Para direccionamiento directo o inmediato hay que acceder a los campos opcionales

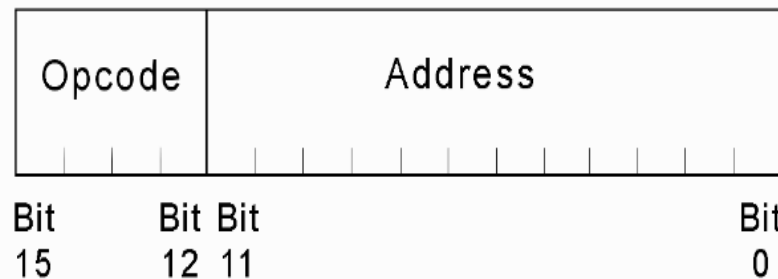
Ejemplo: MARIE

Máquina de Acumulador:

- Representación binaria, complemento a 2.
 - Instrucciones de tamaño fijo.
 - Memoria accedida por palabras de 4K.
 - Palabra de 16 bits.
 - Instrucciones de 16 bits, 4 para el código de operación y 12 para las direcciones.
 - Una ALU de 16 bits.
 - 7 registros para control y movimiento de datos.
- 

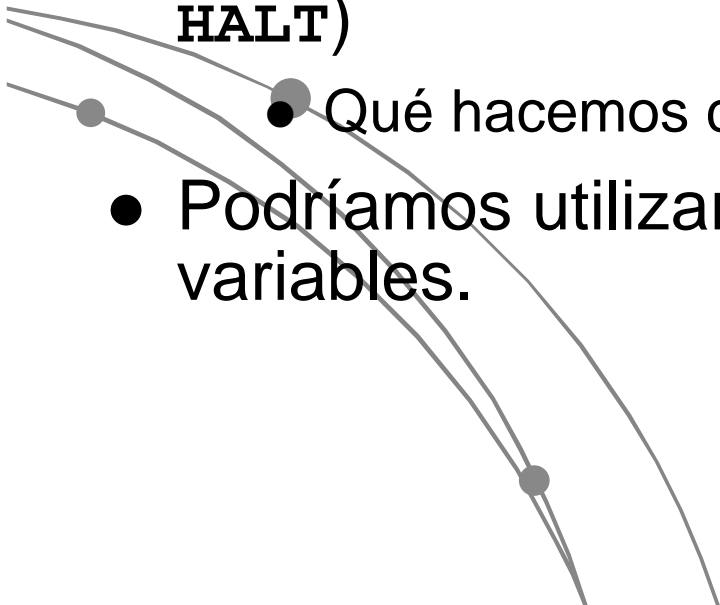
MARIE

- Registros Visibles:
 - AC: Acumulador
 - 16 bits
 - Operando implícito en operaciones binarias
 - También para condiciones.
- Formato de instrucción fijo



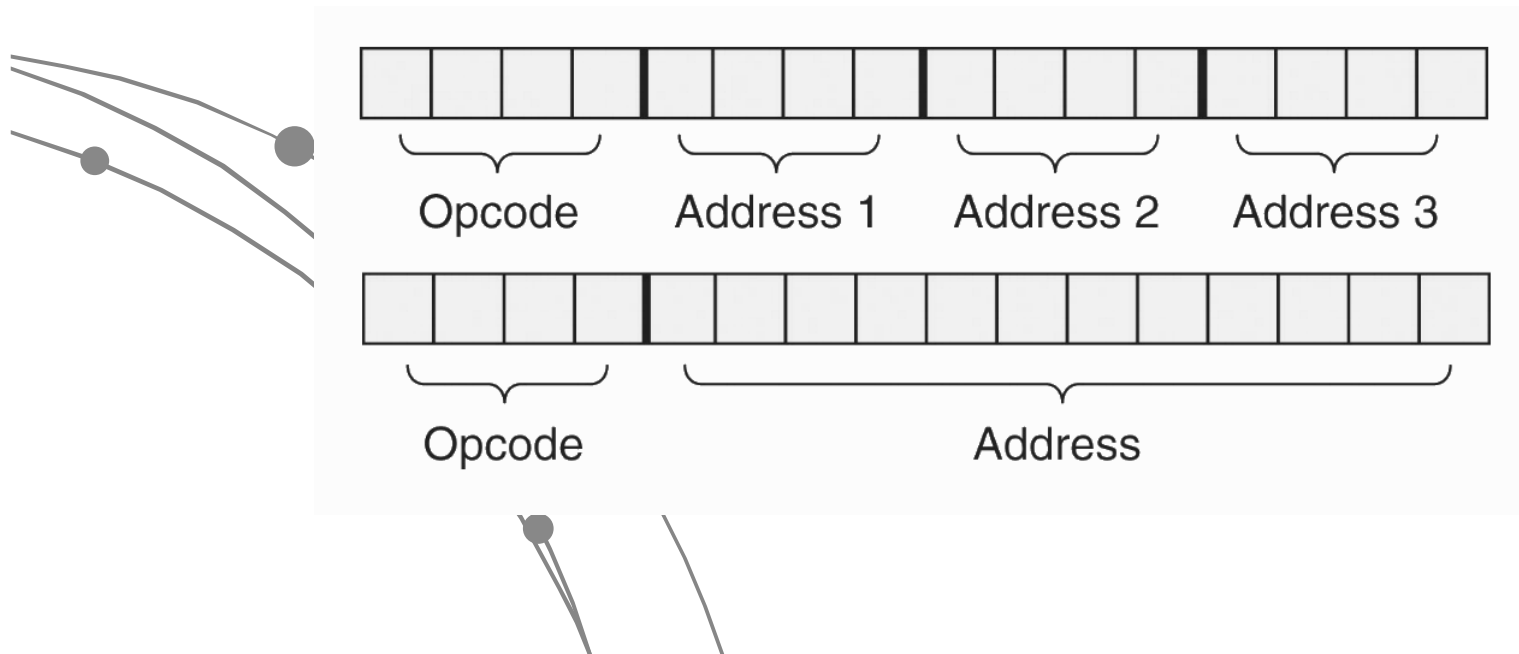
Formatos de Instrucción

- El tamaño de las instrucciones está fuertemente influenciado por el número de operandos que soporta el ISA.
- No todas las instrucciones requieren el mismo número de operandos.
- Hay operaciones que no requieren operandos (ej: **HALT**)
- Qué hacemos con el espacio que sobra?
- Podríamos utilizar códigos de operación variables.



Ejemplo Máquina con Registros

- 16 Registros, 4K de memoria.
- Necesitamos 4 bits para acceder a un registro
- Necesitamos 12 bits para acceder a memoria.
- Si las instrucciones son de 16-bits tenemos dos opciones:



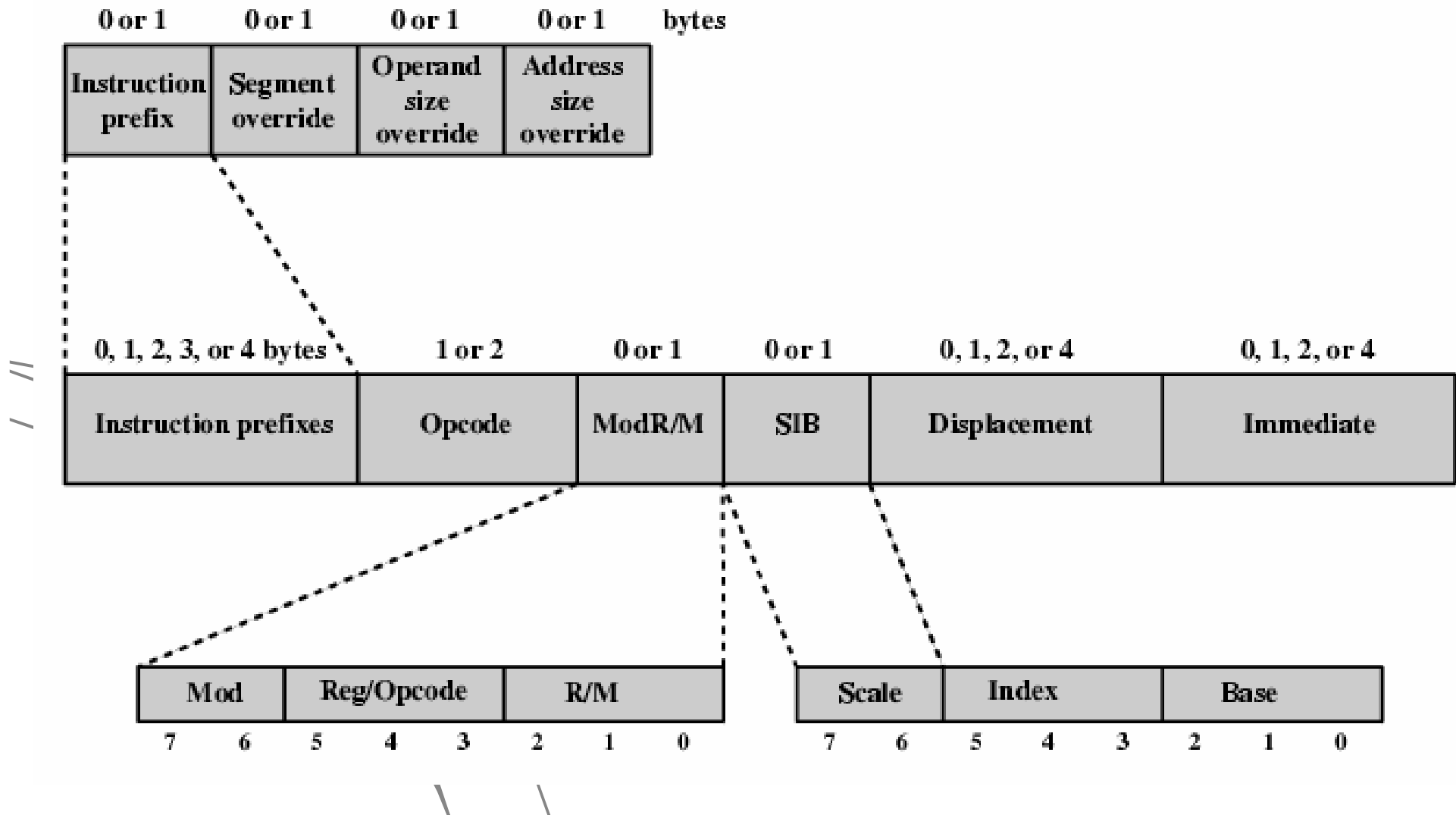
Ejemplo

- Si permitimos que varíe el opcode:

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		

Falta algo?

Formato de Instrucción Pentium



Ejemplo

Prefijo	OpCode	MODR/M	SIB	Desplazamiento	Inmediato
0xF3	0xA4				

Instrucción: REP MOVSB

Copia CX bytes de DS:SI, a ES:DI.

MOVSB: Copia el dato en DS:SI, a ES:DI.

Dependiendo de un flag, SI y DI son incrementados (+1) o decrementados (-1)

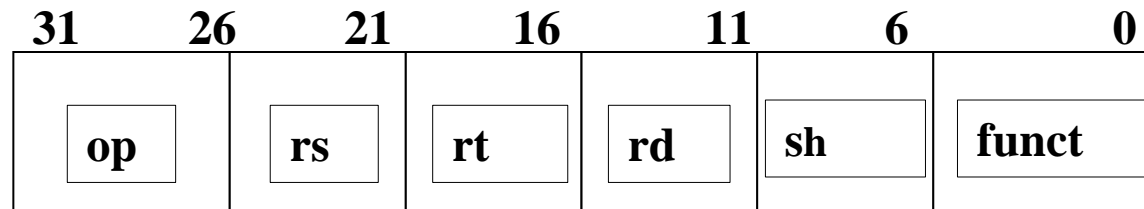
REP decrementa CX y hace que se repita la operación hasta que CX llegue a 0

Formato MIPS de Instrucción

Son todas de 32 bits. Tres formatos:

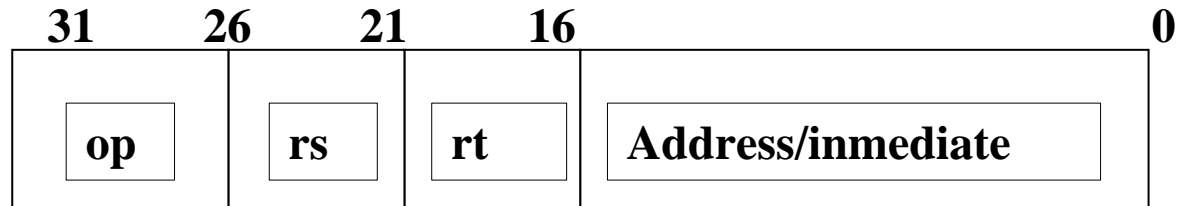
- Tipo R

- Aritméticas



- Tipo I

- Transferencia, salto
- Operaciones con operando inmediato

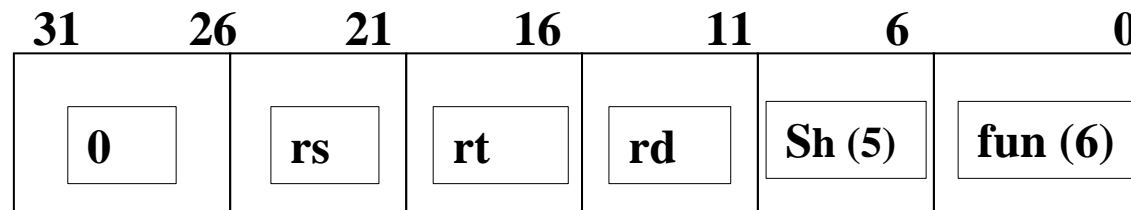


- Tipo J

- Saltos



Formato R (registro)



- Op= 0
- rs, rt = identificación de los registros fuente
- rd = identificación del registro destino
- sh= cantidad de shifts
- func= identifica la operación (por ej. add=32, sub=34, sll=0,srl=10)

Formato R :Ejemplos

- Add \$1, \$2, \$3

31	26	21	16	11	6	0
0	2	3	1	0	32	

- Sub \$1, \$2, \$3

31	26	21	16	11	6	0
0	2	3	1	0	34	

- Slt \$1, \$2, \$3

31	26	21	16	11	6	0
0	2	3	1	0	42	

Set Less Than

si ($\$2 < \3) entonces $\$1 = 1$ sino $\$1 = 0$

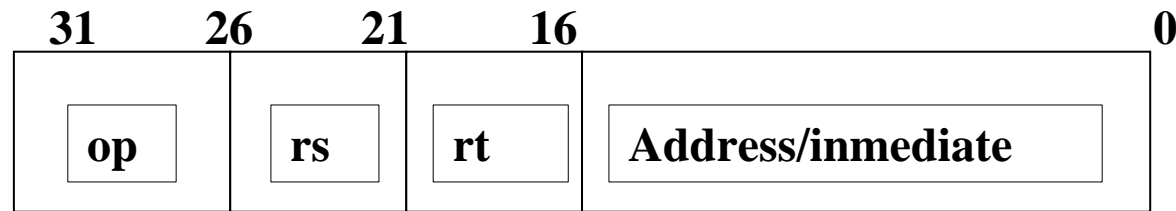
- Jr \$31

31	26	21	16	11	6	0
0	31	0	0	0	8	

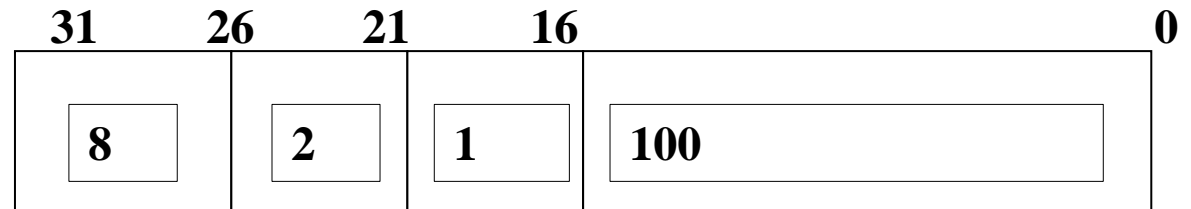
Jump Register

$PC \leftarrow \$31$

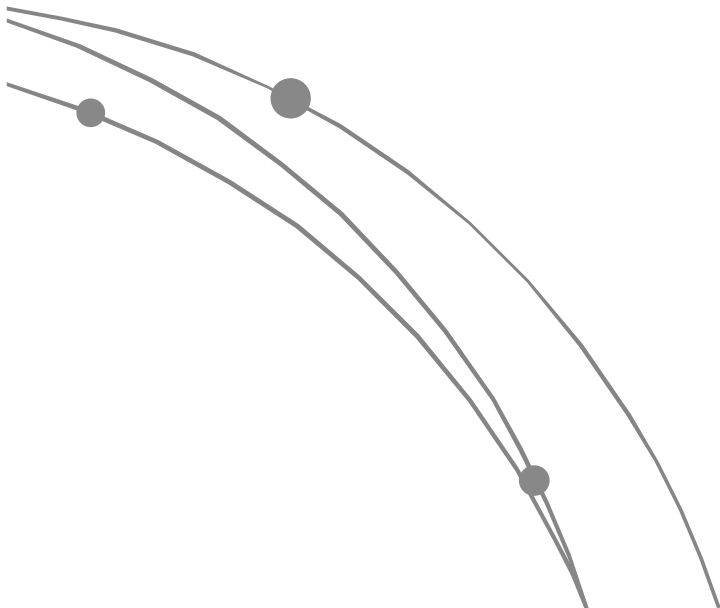
Formato I : Transferencias inmediatas



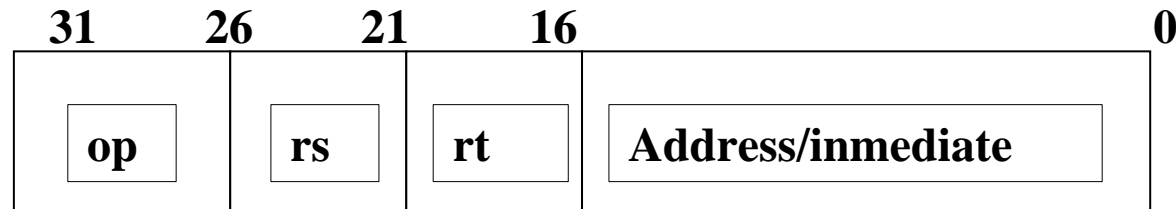
- Addi \$1,\$2,100



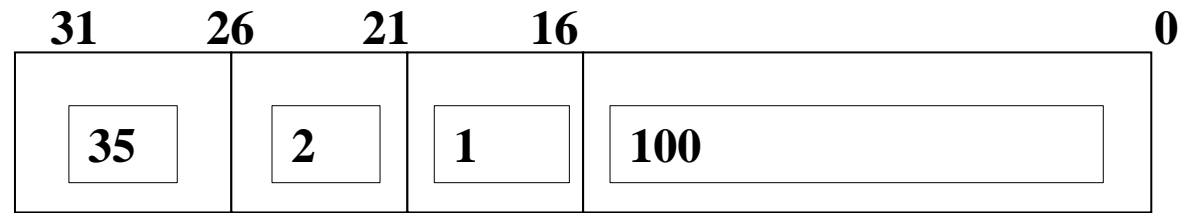
$\$1 = \$2 + 100$



Formato I : Transferencias

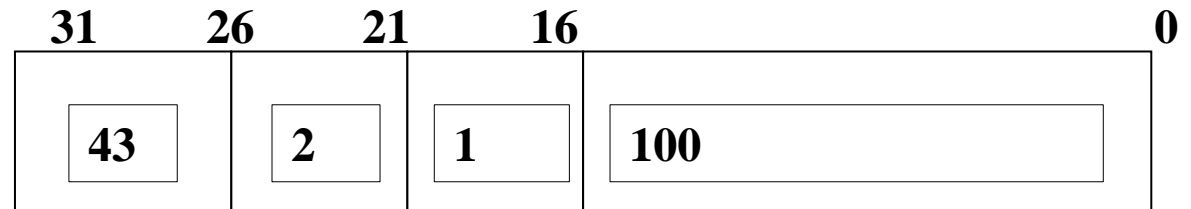


- Lw \$1, 100(\$2)



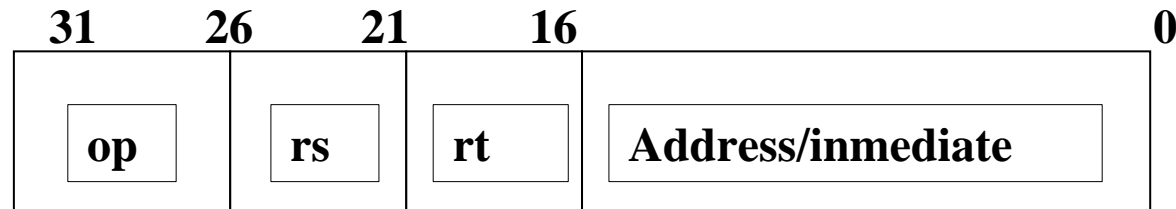
Load Word $\$1 = M[\$2 + 100]$

- Sw \$1, 100(\$2)



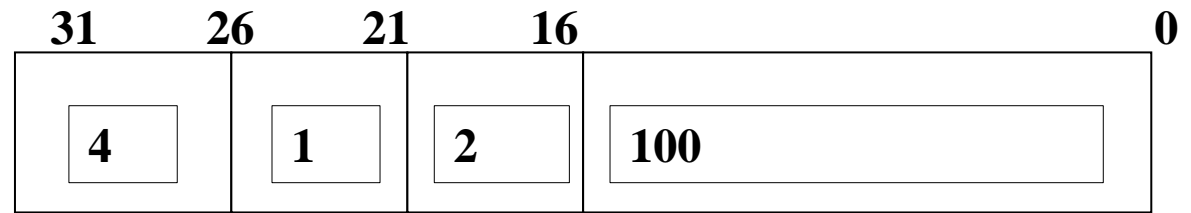
Store Word $M[\$2 + 100] = \1

Formato I : Saltos Inmediatos



- Beq\$1,\$2,100

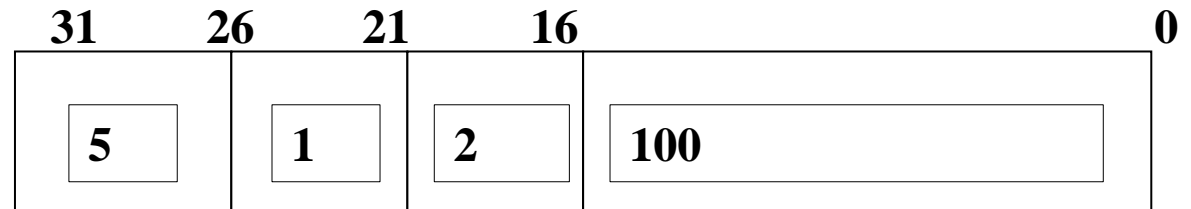
Branch Equal



si (\$1=\$2) entonces ir a PC+4+100

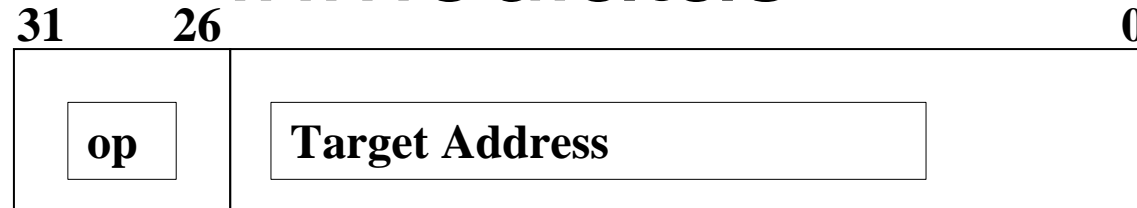
- Bne\$1,\$2,100

Branch Not Equal

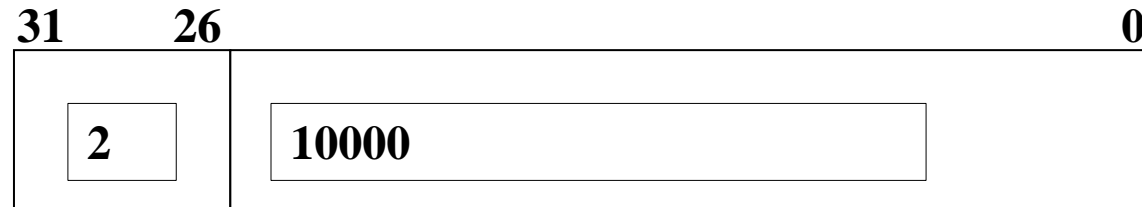


si (\$1!= \$2) entonces ir a PC+4+100

Formato J : Transferencias inmediatas



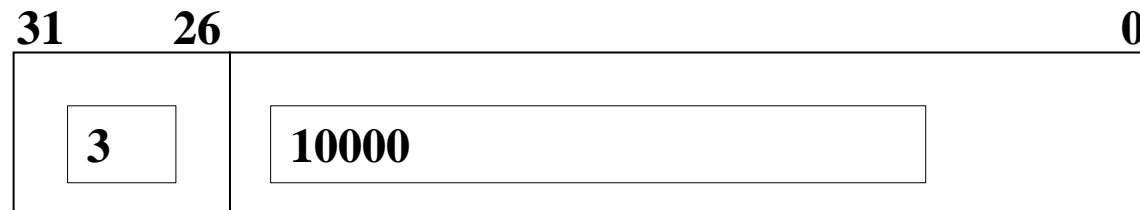
• J 10000



$PC_{27..2}=10000$

$PC_{1..0}=00$

• Jal 10000

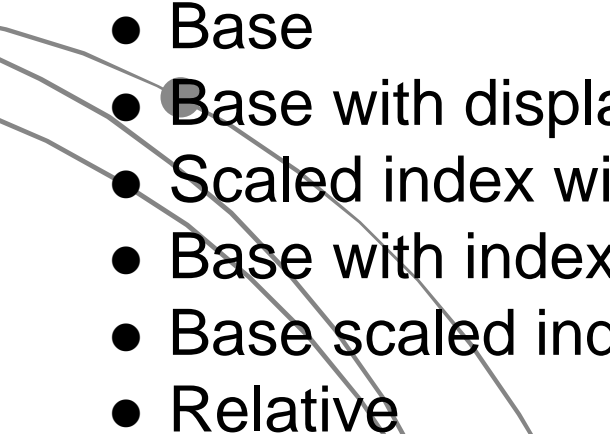


JUMP AND LINK $\$31=PC+4$

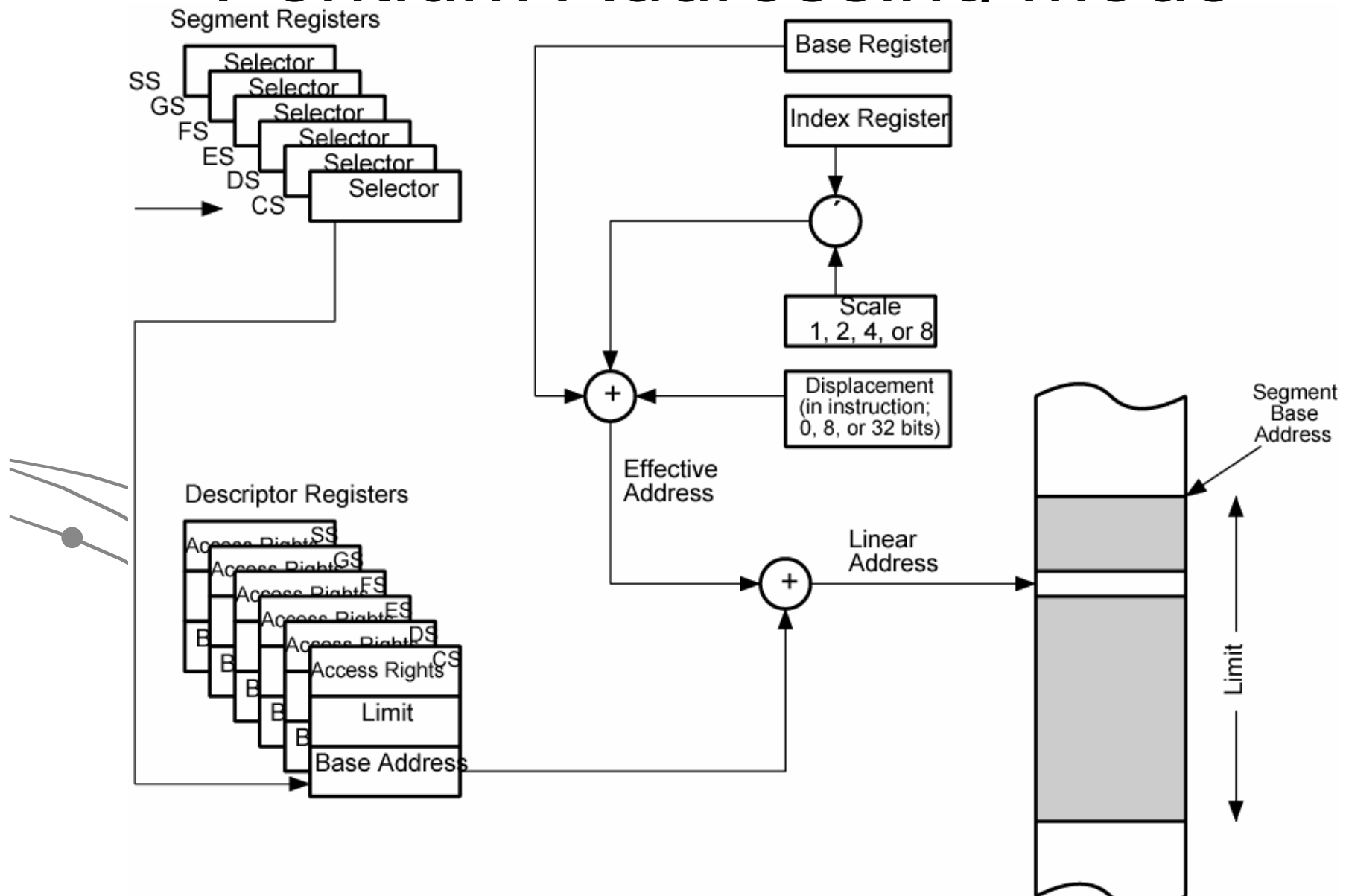
$PC_{27..2}=10000$

$PC_{1..0}=00$

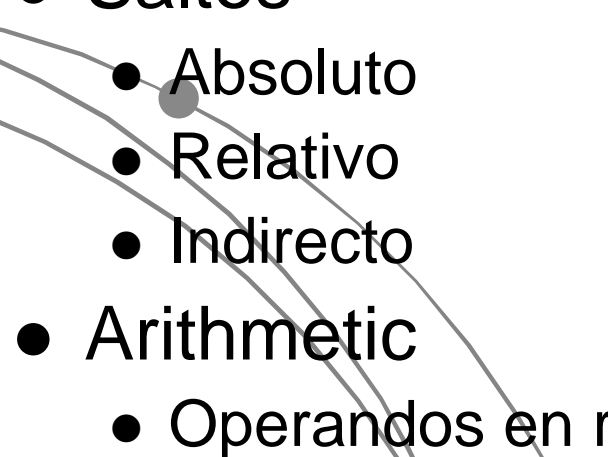
Pentium Addressing Modes

- Direccional usando Segmento + Offset
 - Segmento + Offset = Dirección Plana
 - 12 modos disponibles
 - Immediate
 - Register operand
 - Displacement
 - Base
 - Base with displacement
 - Scaled index with displacement
 - Base with index and displacement
 - Base scaled index with displacement
 - Relative
- 
- A decorative graphic consisting of two curved lines, one above the other, both starting from the left and curving downwards towards the right. Three small grey dots are placed along these lines: one on the upper line, one on the lower line, and one on the lower line further to the right.

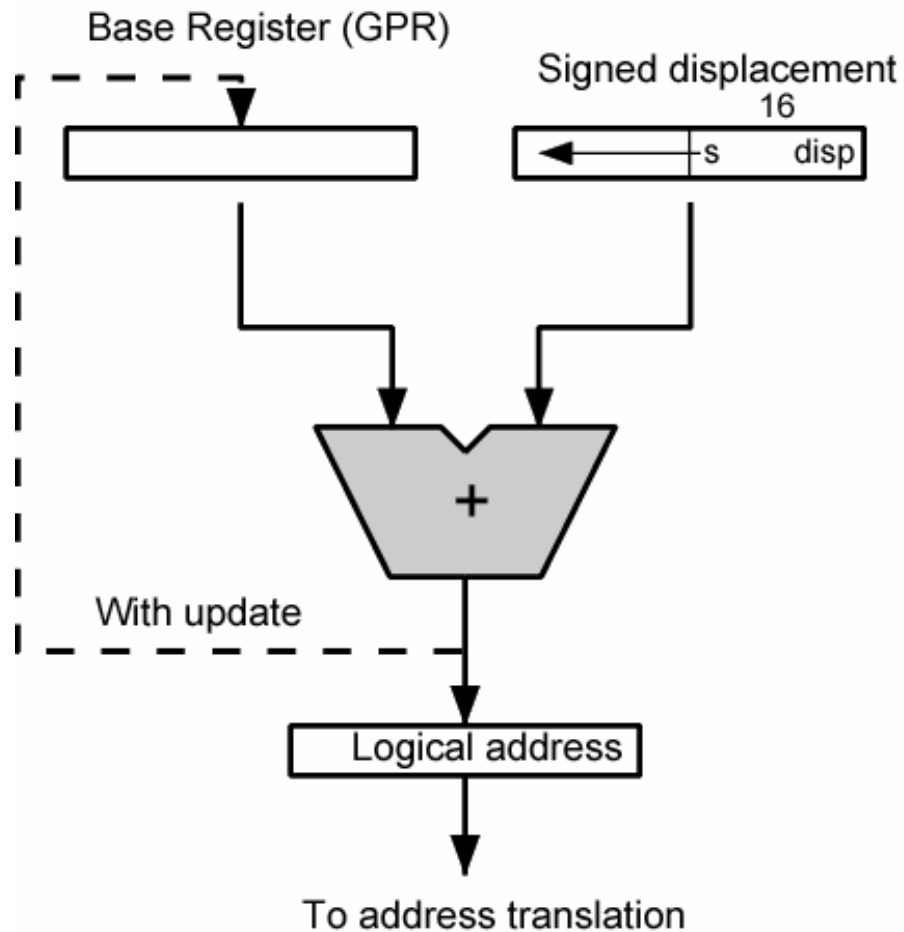
Pentium Addressing Mode



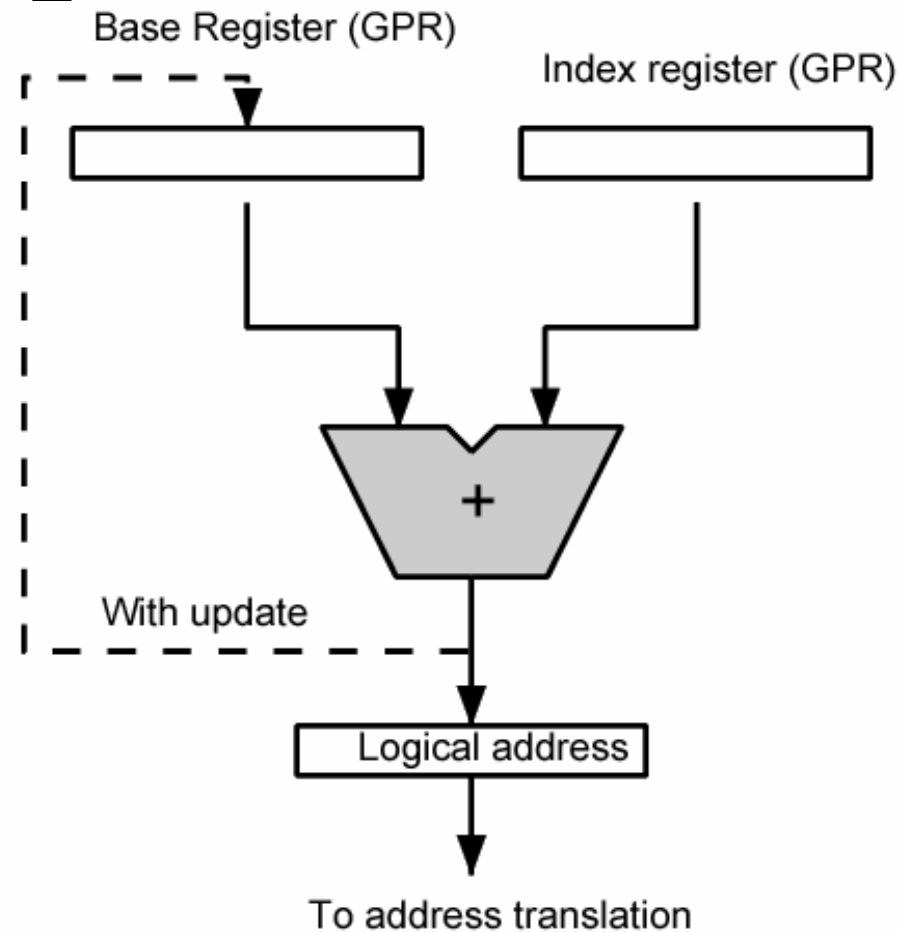
PowerPC Addressing Modes

- Load/store
 - Register Indirect + Desplazamiento
 - Las instrucciones incluyen 16 bit de desplazamiento a ser sumado a un registro base (seteable)
 - Indirect indexed
 - Un registro de base y otro de indice
 - Saltos
 - Absoluto
 - Relativo
 - Indirecto
 - Arithmetic
 - Operandos en registros o en la instrucción
- 

PowerPC Memory Operand Addressing Modes



(a) Indirect Addressing



(b) Indirect Indexed Addressing

Referencias

- Tanenbaum – Capitulo 5
- Stalling – Capitulo 11
- Null - Capitulo 5

