



# TEMA VI

## DISEÑO DEL PROCESADOR

## **Diseño del procesador**

### **6.1 Repertorio de instrucciones**

- 6.1.1 Procesadores de tres direcciones
- 6.1.2 Procesadores de dos direcciones
- 6.1.3 Procesadores de una dirección (procesadores con acumulador)
- 6.1.4 Procesadores de cero direcciones (procesadores con pila)
- 6.1.5 Procesadores sin ALU
- 6.1.6 Análisis de las diferentes arquitecturas de procesadores
- 6.1.7 Procesadores con banco de registros
- 6.1.8 Arquitectura de carga/almacenamiento: Procesadores RISC

### **6.2 Modos de direccionamiento**

### **6.3 Ciclo de ejecución de una instrucción**

- 6.3.1 Fase de búsqueda de la instrucción
- 6.3.2 Fase de decodificación de la instrucción
- 6.3.3 Fase de búsqueda de los operandos
- 6.3.4 Fase de ejecución de la instrucción
- 6.3.5 Transferencia a un subprograma
- 6.3.6 Ciclo de interrupción

### **6.4 Fases en el diseño del procesador**

### **6.5 Diseño de un procesador elemental**

- 6.5.1 Especificación del procesador SIMPLE1
- 6.5.2 Repertorio de instrucciones
- 6.5.3 Diagrama de flujo del repertorio de instrucciones
- 6.5.4 Asignación de recursos a la unidad de procesamiento o ruta de datos
- 6.5.5 Obtención del diagrama ASM del procesador
- 6.5.6 Diseño de la unidad de control
- 6.5.7 Diseño de la unidad de procesamiento o ruta de datos

## 6.2 Modos de direccionamiento

- El modo de direccionamiento determina la forma que se interpreta el contenido del campo de dirección de una instrucción
- El modo de direccionamiento permite calcular de forma no ambigua la dirección real donde se encuentran los operandos
- Las ventajas:
  - Reducción del tamaño de las instrucciones
  - Aumento de la flexibilidad de la programación
- Modos de direccionamiento
  - Implícito
  - Inmediato
  - Directo
  - Relativo
  - Indirecto
  - Indexado

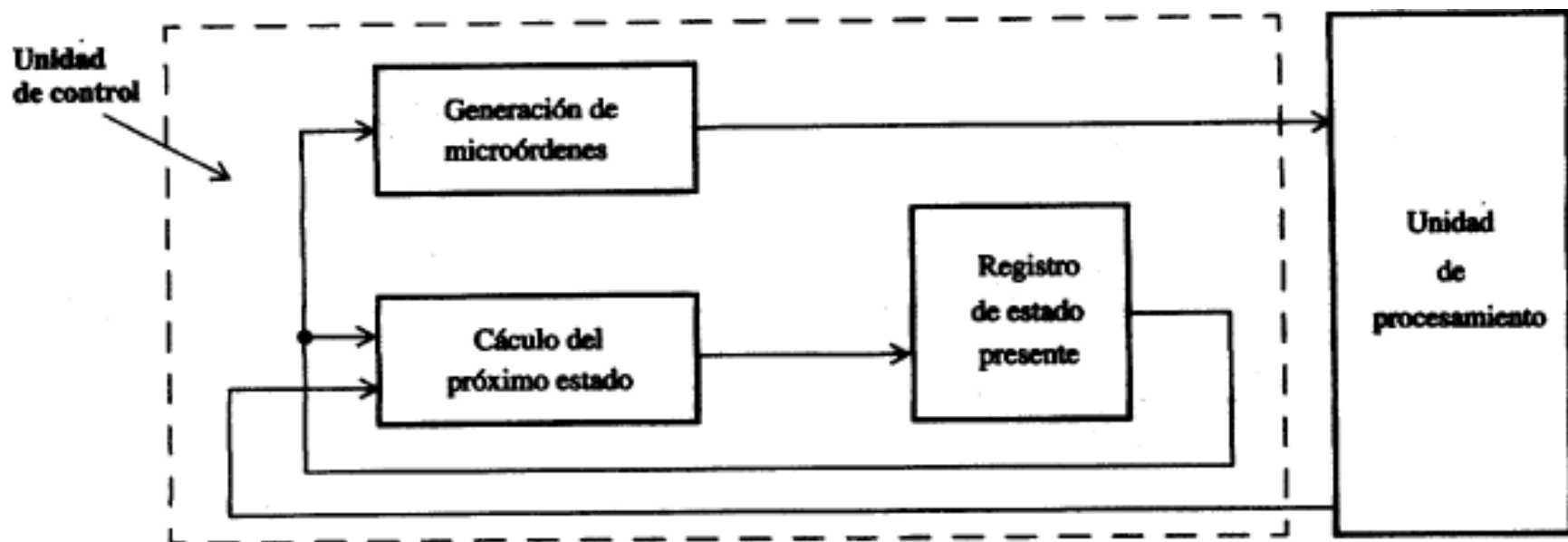
# DISEÑO DEL PROCESADOR

## ■ Partes de un sistema digital

- Unidad de procesamiento:
  - Se almacenan y transforman los datos
- Unidad de control:
  - Genera las secuencias de señales de control de acuerdo al algoritmo de transferencia de registros.

## ■ Tipos de microoperaciones

- De transferencia
- De proceso



**Figura 6.2:** Estructura de una unidad de control con lógica cableada

Unidad de control

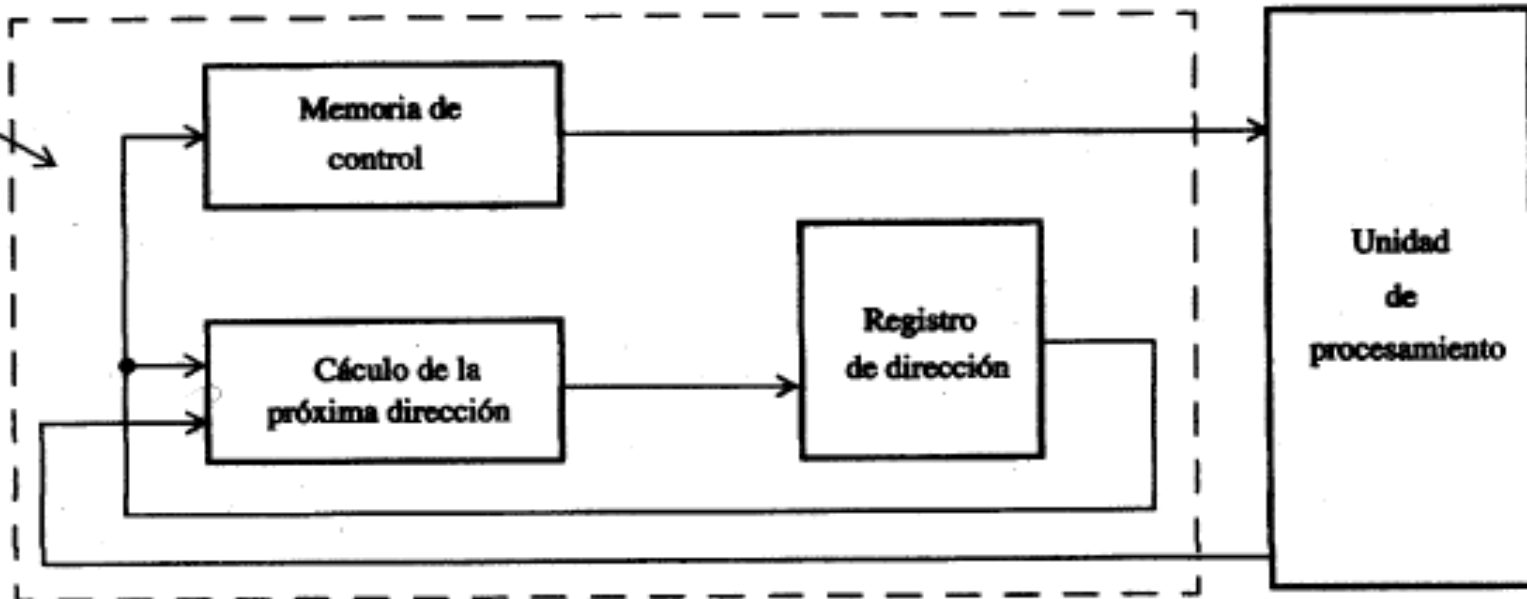


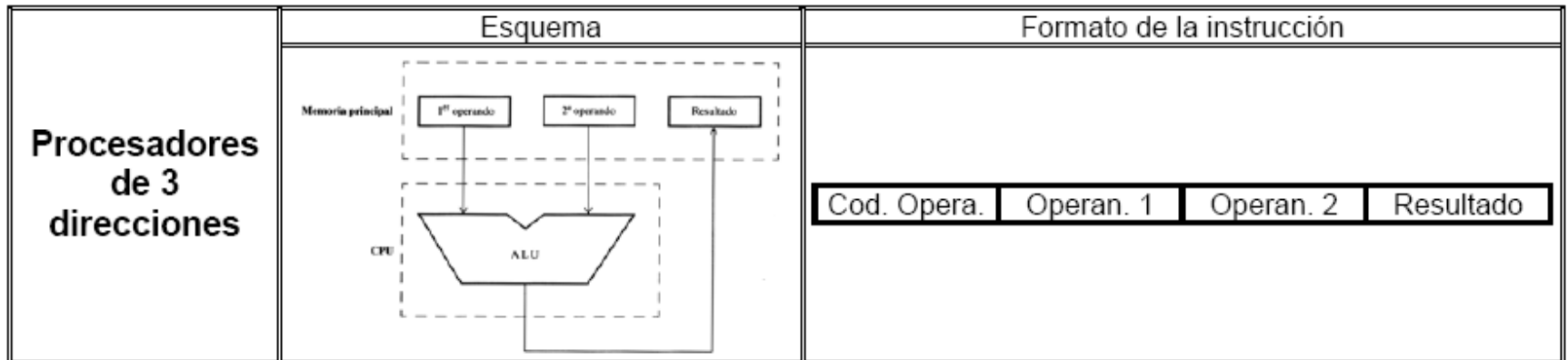
Figura 6.3: Estructura de una unidad de control microprogramada

## 6.1 Repertorio de instrucciones

- Cada CPU tiene su propio y específico formato de instrucciones
- Una instrucción una cadena de bits que se agrupan en campos con tamaños diferentes
- Tipos de instrucciones
  - De transferencia de datos
  - Aritméticas, lógicas y de comparación
  - De desplazamiento
  - De Transferencia de control
  - De gobierno
- **Tipos de procesadores según el número de direcciones**
  - 3 direcciones
  - 2 direcciones
  - 1 dirección
  - 0 direcciones

# 6.1.1 Procesadores de tres direcciones

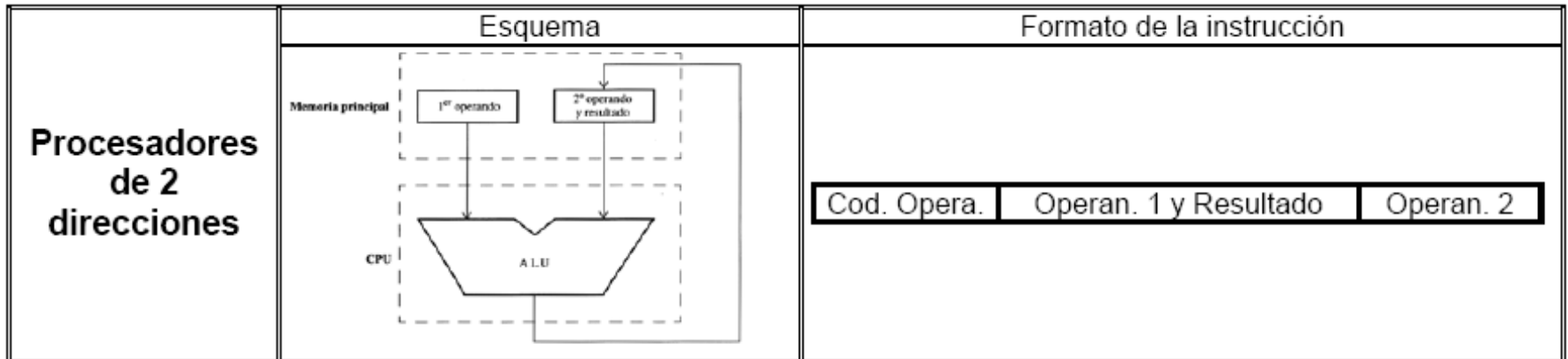
- Requiere un número elevado de bits para codificarla
- Programas cortos



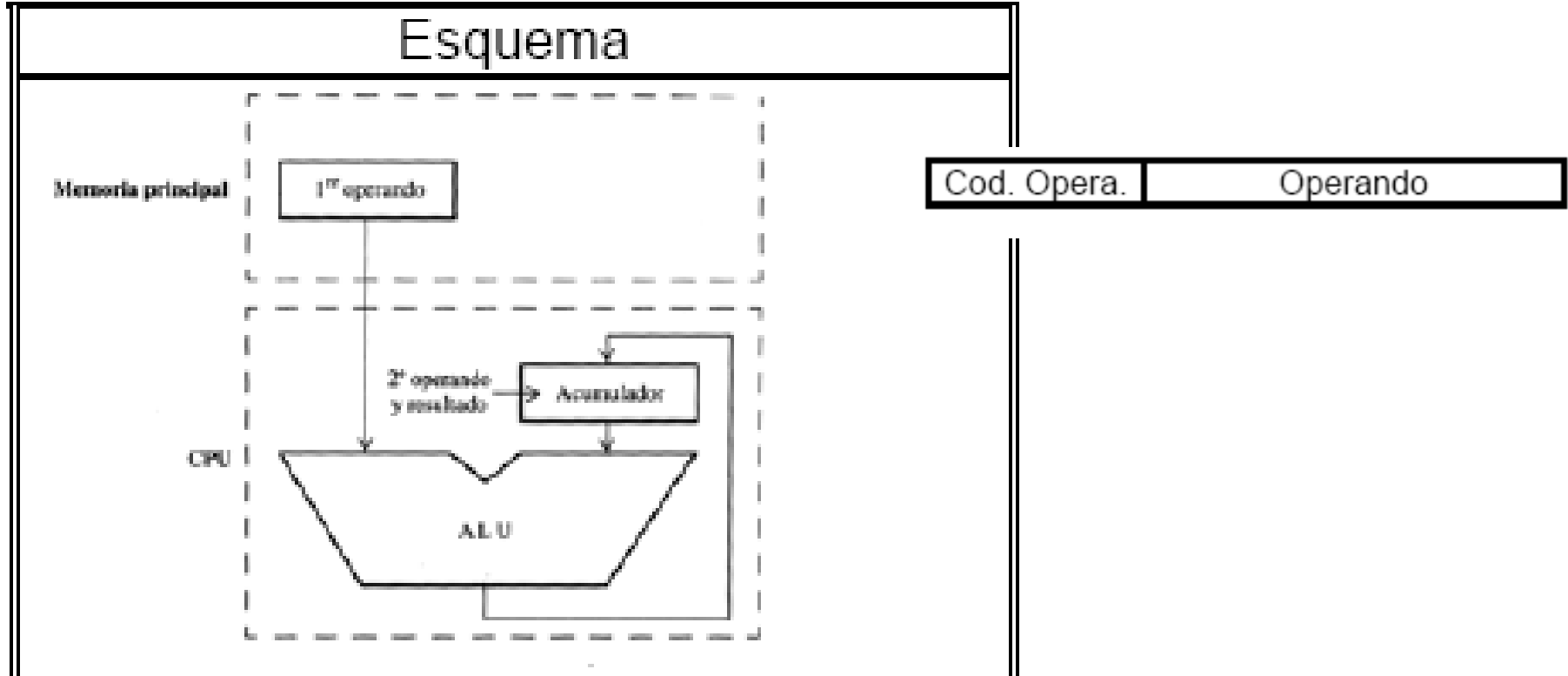


## 6.1.2 Procesadores de dos direcciones

- Mayor longitud del programa
- Menor número de acceso a memoria que necesitan las instrucciones



## 6.1.3 Procesadores de una dirección (procesadores con acumulador)

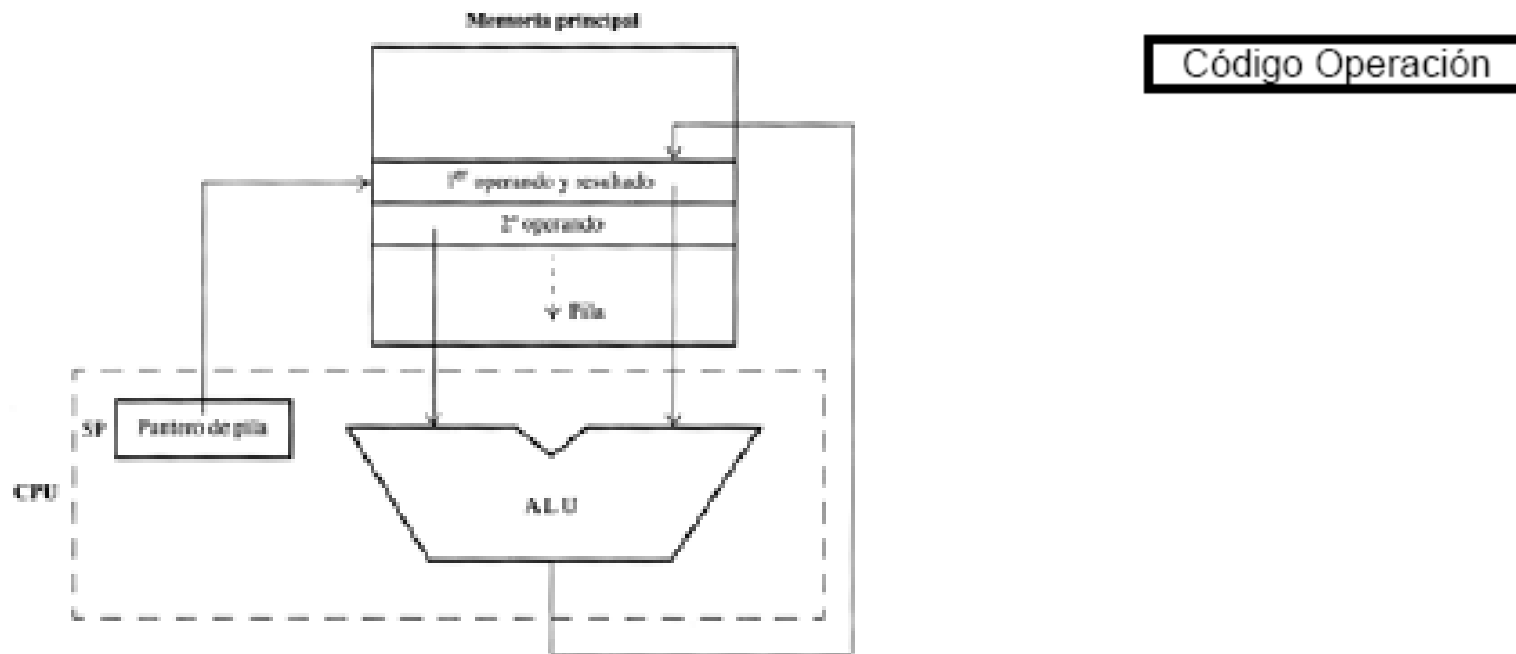


- El procesador dedica un registro como operando destino

## 6.1.4 Procesadores de cero direcciones (procesadores con pila)

- Primer, segundo operando y resultado en la pila

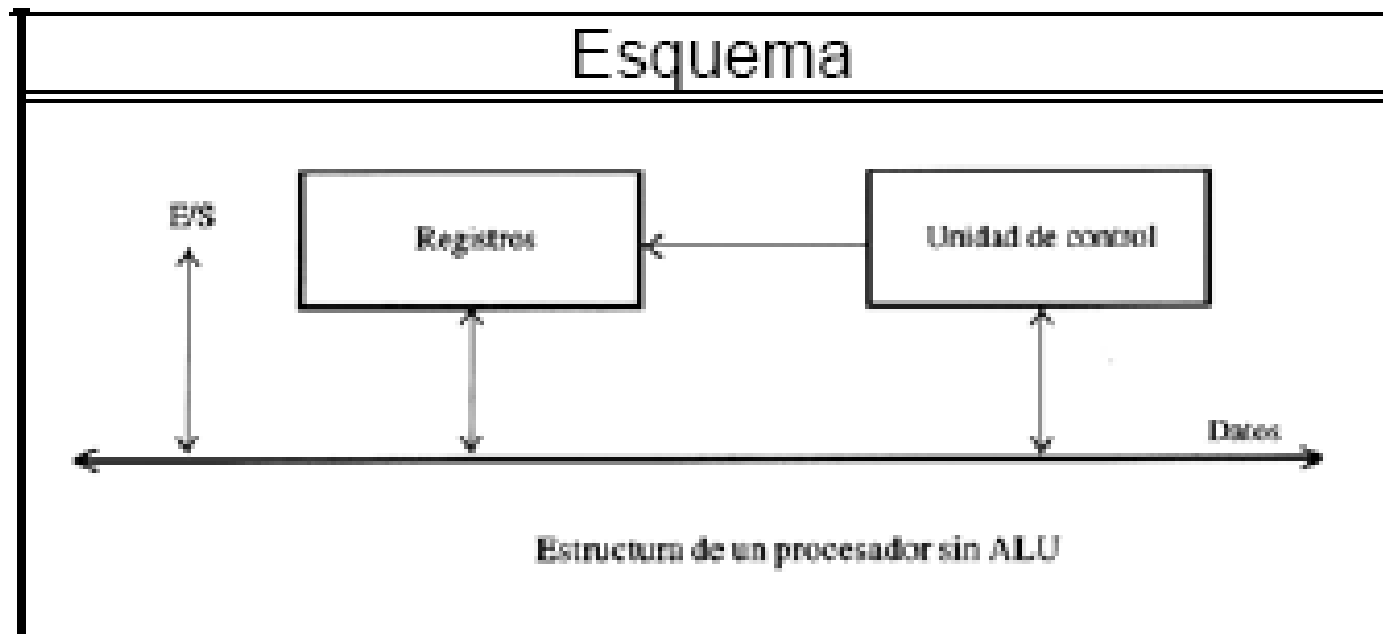
### Esquema



Flujo de datos en un procesador de pila en las instrucciones con referencia a memoria

## 6.1.5 Procesadores sin ALU

- Operaciones de transferencia



## 6.1.6 Análisis de las diferentes arquitecturas de procesadores

- Se han analizados instrucciones ternarias:
  - Dos operandos y un resultado
- El número de instrucciones aumenta conforme disminuye en número de operandos explícitos en cada instrucción

## 6.1.7 Procesadores con banco de registros

- La idea del procesador con acumulador puede generalizarse incrementando el número de registros (banco de registros)
- Ventajas:
  - Al almacenar los resultados intermedios en registros en vez de en memoria principal aumenta la velocidad
  - Menor tamaño de las instrucciones.
    - En vez de una dirección de memoria, en la instrucción aparece el número de registro (se necesitan menos bits para direccionarlos)

## 6.1.8 Arquitectura de carga/almacenamiento: Procesadores RISC

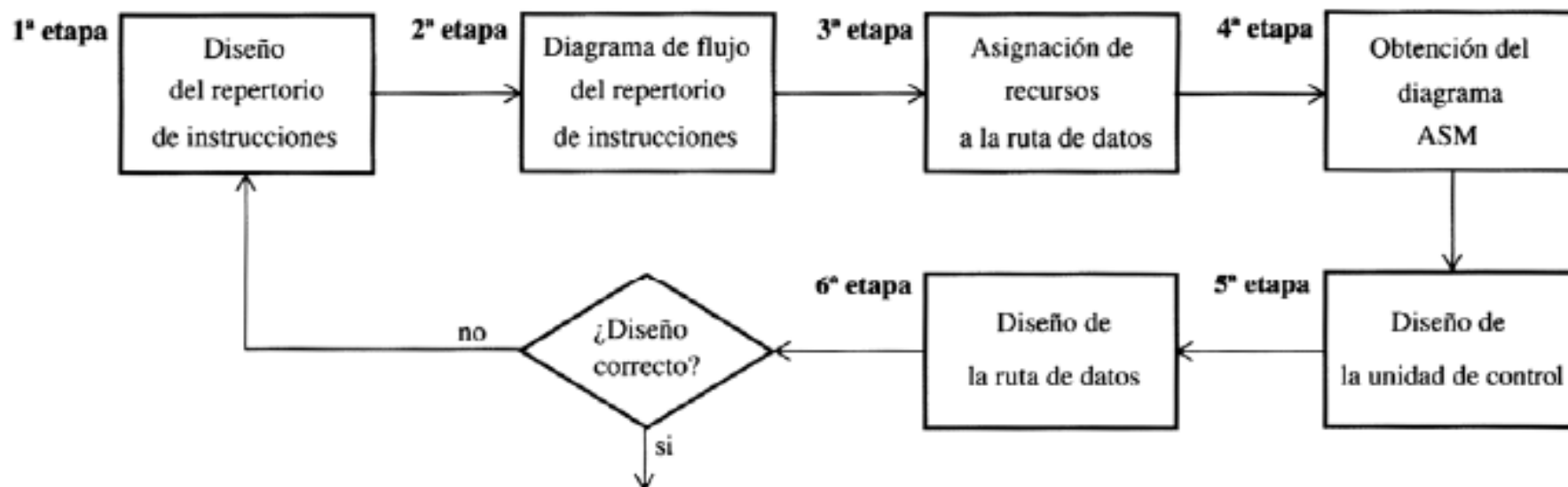
- RISC (Reduce Instruction Set Computer)
  - Arquitectura Carga/almacenamiento
    - Se accede a memoria solo para extraer datos o poner resultados
    - El resto de las operaciones se realizan en los registros
  - Instrucciones sencillas (operaciones elementales)
  - Formato de instrucciones regular (misma longitud)
  - Unidad de control cableada y ciclo por instrucción suele ser uno
  - Modo de direccionamientos limitados
- CISC (Complex Instruction Set Computer)

## 6.3 Ciclo de ejecución de una instrucción

- 6.3.1 Fase de búsqueda de la instrucción
- 6.3.2 Fase de decodificación de la instrucción
- 6.3.3 Fase de búsqueda de los operandos
- 6.3.4 Fase de ejecución de la instrucción
- 6.3.5 Transferencia a un subprograma
- 6.3.6 Ciclo de interrupción



## 6.4 Fases en el diseño del procesador



Etapas en el diseño del procesador

# 6.5 Diseño de un procesador elemental

## ■ 6.5.1 Especificación del procesador SIMPLE1

□ SIMPLE1 ordenador elemental académico

□ Registros

■ Registro contador de programa (PC): Como la capacidad del operando → 9 bits

■ Reg. Instrucciones (IR): Anchura del formato de la instrucción → 12 bits

■ Reg. Direcc. Mem. (MAR): Capacidad de direccionamiento → 9 bits

■ Reg. Datos de memoria (MBR): Anchura del formato de la instruc. → 12 bits

■ Registros de trabajo A , B: Como la capacidad del operando → 9 bits

■ ALU: Suma / resta

□ Formato de instrucción:

11	10	9	8	7	6	5	4	3	2	1	0
Cod. Operac.			Operando								

## 6.5.2 Repertorio de instrucciones

Nemotécnico	Código binario	Instrucción	Acción
LDA x	LDA = 001	Carga directa	$A \leftarrow M[x]$
STA x	STA = 010	Almacenamiento directo	$M[x] \leftarrow A$
ADD	ADD = 011	Suma B a A	$A \leftarrow A + B$
SUB	SUB = 100	Resta B de A	$A \leftarrow A - B$
MAB	MAB = 101	Mueve A a B	$B \leftarrow A$
BR x	BR = 110	Salto incondicional a x	$PC \leftarrow x$
BRN x	BRN = 111	Salto a x si indicador negativo a 1	$PC \leftarrow x$ si $IN = 1$

Repertorio de instrucciones de SIMPLE1

## 6.5.3 Diagrama de flujo del repertorio de instrucciones

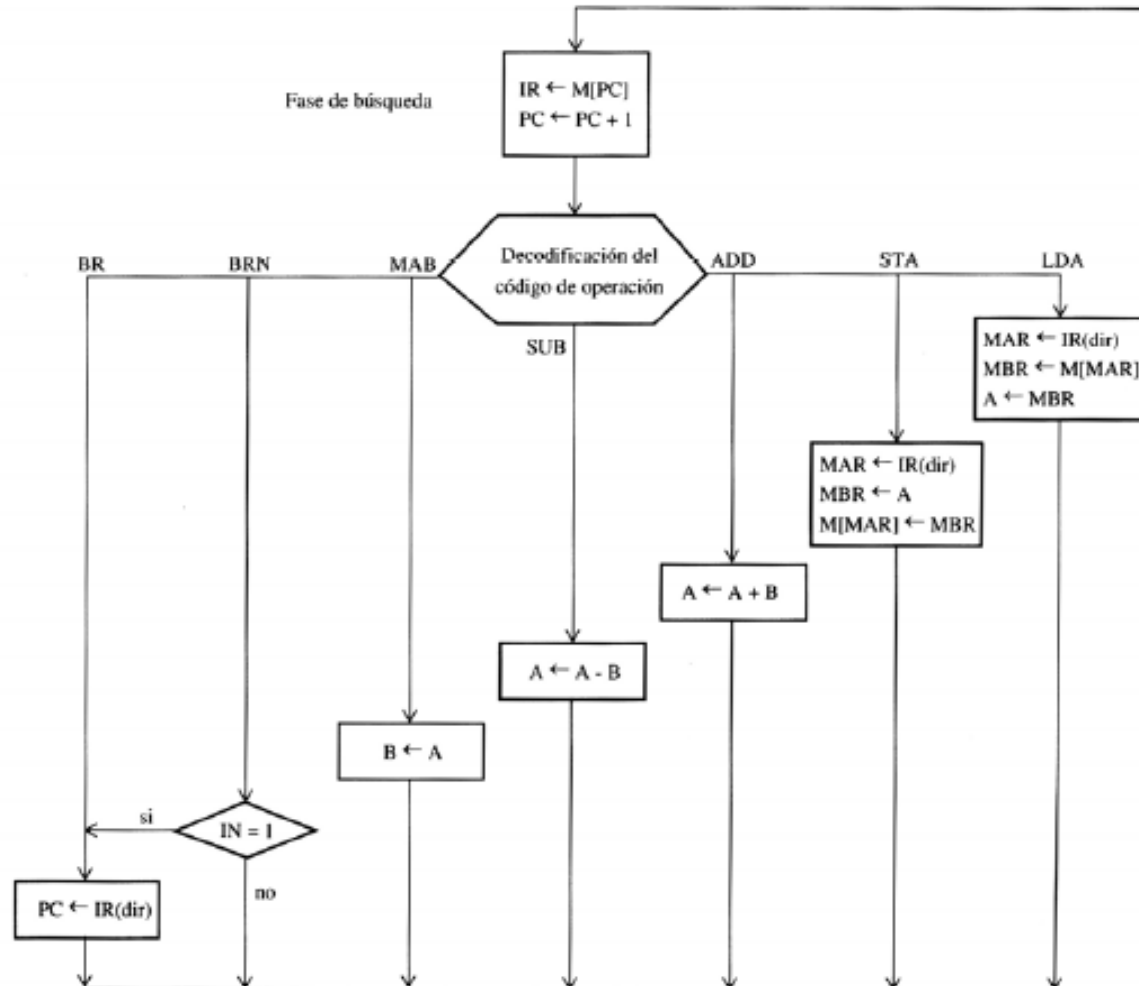
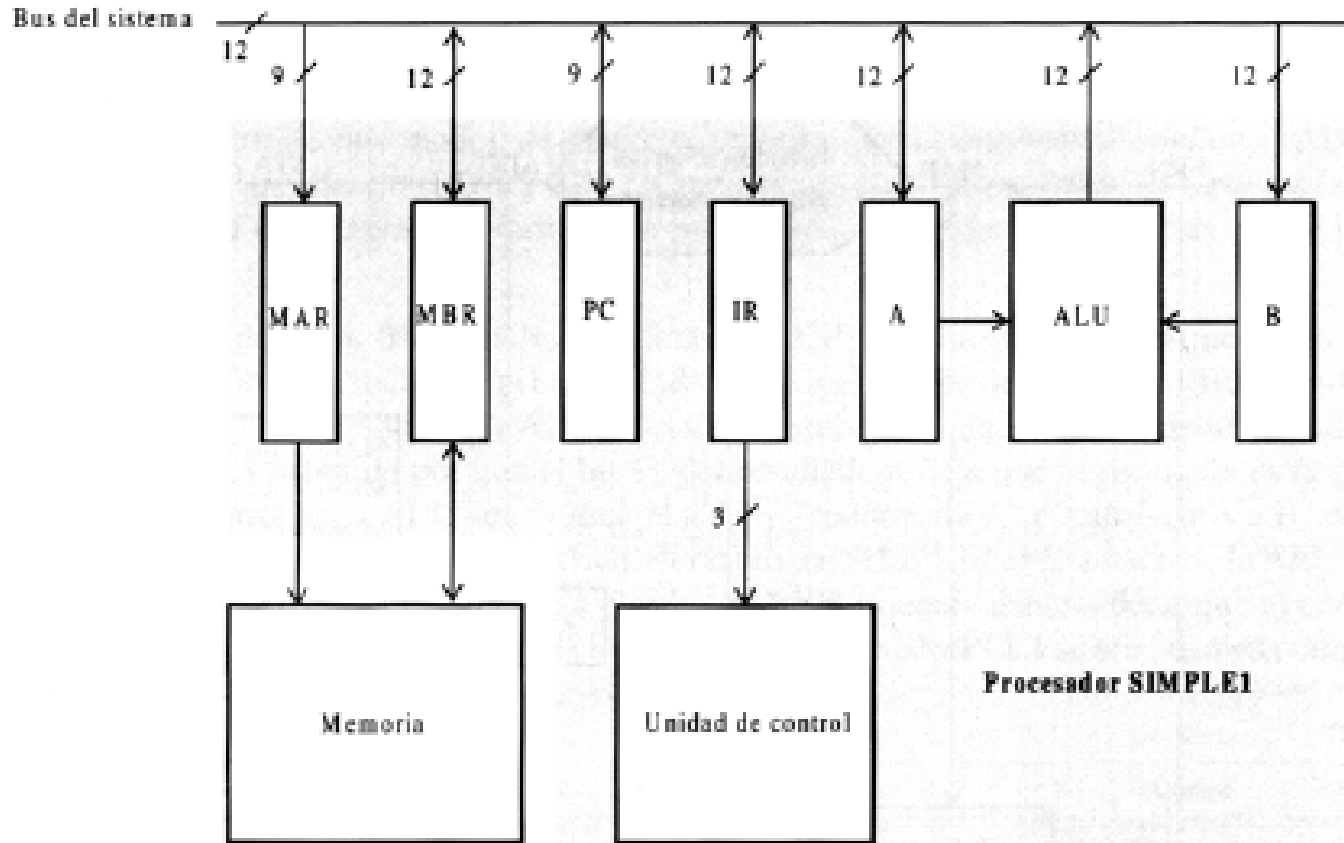


Diagrama de flujo del repertorio de instrucciones de SIMPLE1

## 6.5.4 Asignación de recursos a la unidad de procesamiento o ruta de datos



NO SE HAN INCLUIDO LAS SEÑALES DE CONTROL Y CONDICIÓN

# 6.5.5 Obtención del diagrama ASM del procesador

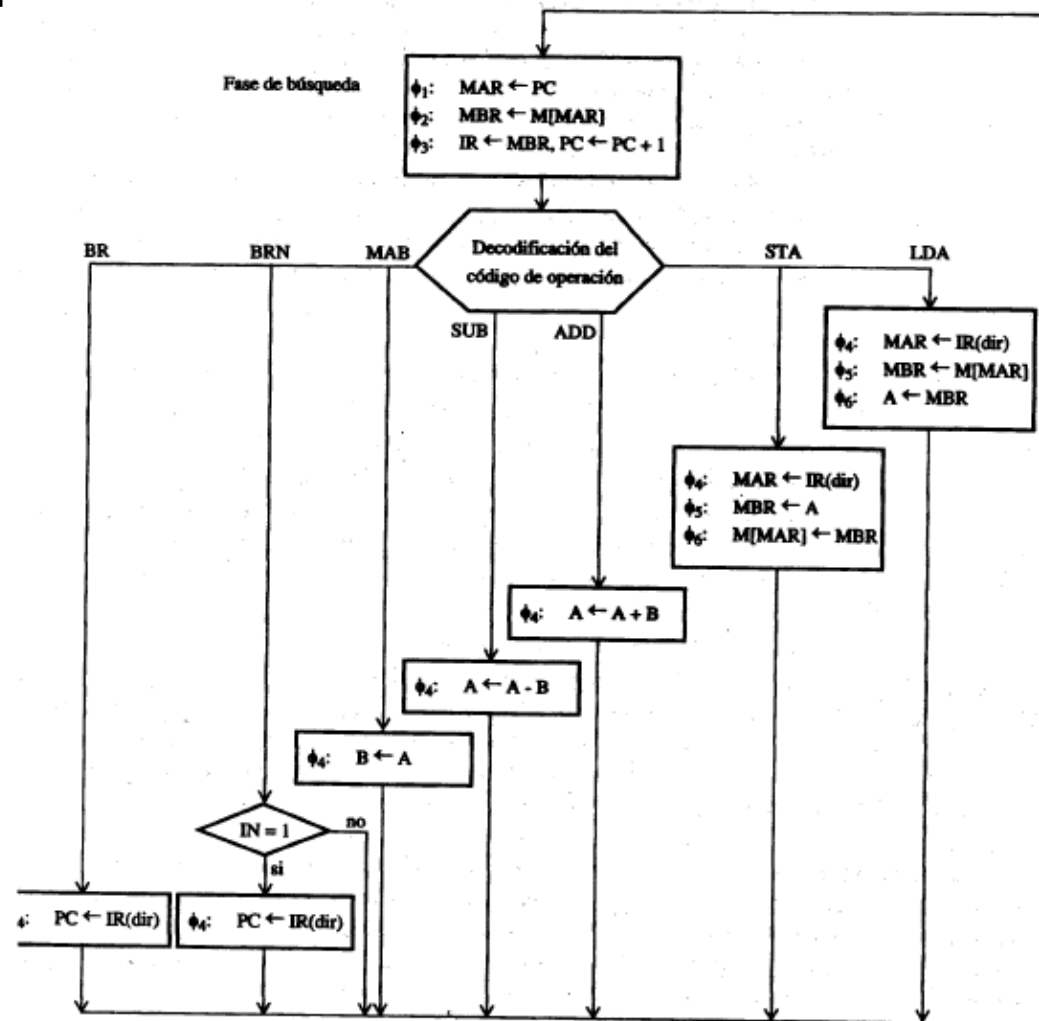
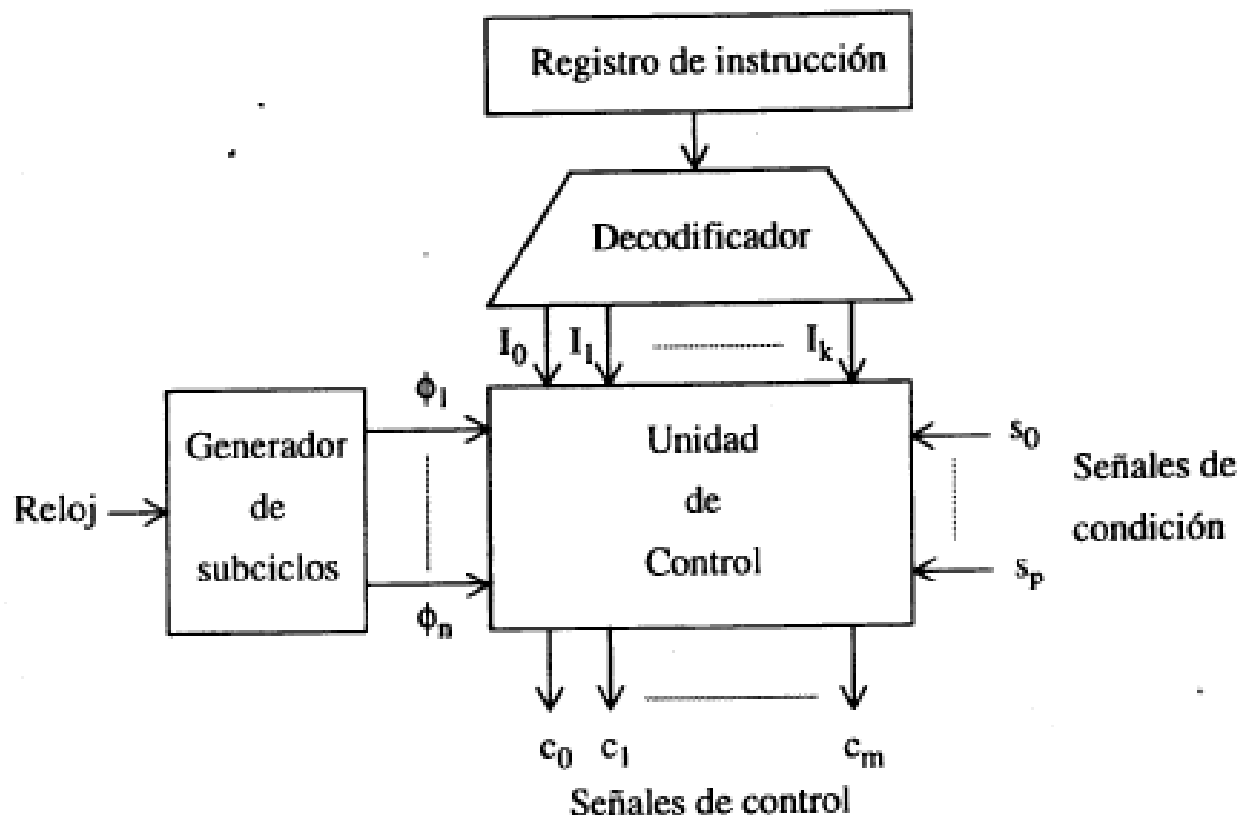


Figura 6.43: Diagrama ASM del procesador SIMPLE1

## 6.5.6 Diseño de la unidad de control



Entradas IR, Señales de condición y Reloj

IR: Necesita un decodificador

Reloj: Dividir el ciclo de una instrucción en subciclos

# Señales de control del SIMPLE1

Señal de control	Microorden controlada
R	Leer de la memoria ( $MBR \leftarrow M[MAR]$ )
W	Escribir en la memoria ( $M[MAR] \leftarrow MBR$ )
CMAR	Cargar el contenido del bus en MAR ( $MAR \leftarrow Bus$ )
HMBR	Habilitar el registro MBR ( $Bus \leftarrow MBR$ )
CMBR	Cargar el contenido del bus en MBR ( $MBR \leftarrow Bus$ )
HPC	Habilitar el registro PC ( $Bus \leftarrow PC$ )
CPC	Cargar el contenido del bus en PC ( $PC \leftarrow Bus$ )
IPC	Incrementar el contenido de PC ( $PC \leftarrow PC + 1$ )
HIR	Habilitar el registro IR ( $Bus \leftarrow IR$ )
CIR	Cargar el contenido del bus en IR ( $IR \leftarrow Bus$ )
HA	Habilitar el registro A ( $Bus \leftarrow A$ )
CA	Cargar el contenido del bus en A ( $A \leftarrow Bus$ )
CB	Cargar el contenido del bus en B ( $B \leftarrow Bus$ )
HALU	Habilitar la unidad aritmético-lógica
SUMA	Seleccionar la función de suma en la unidad aritmético-lógica
RESTA	Seleccionar la función de resta en la unidad aritmético-lógica

H	Habilitar salida de registro a bus
C	Cargar registro desde bus

Señales de control del procesador SIMPLE1



# Unidad de Control del SIMPLE1

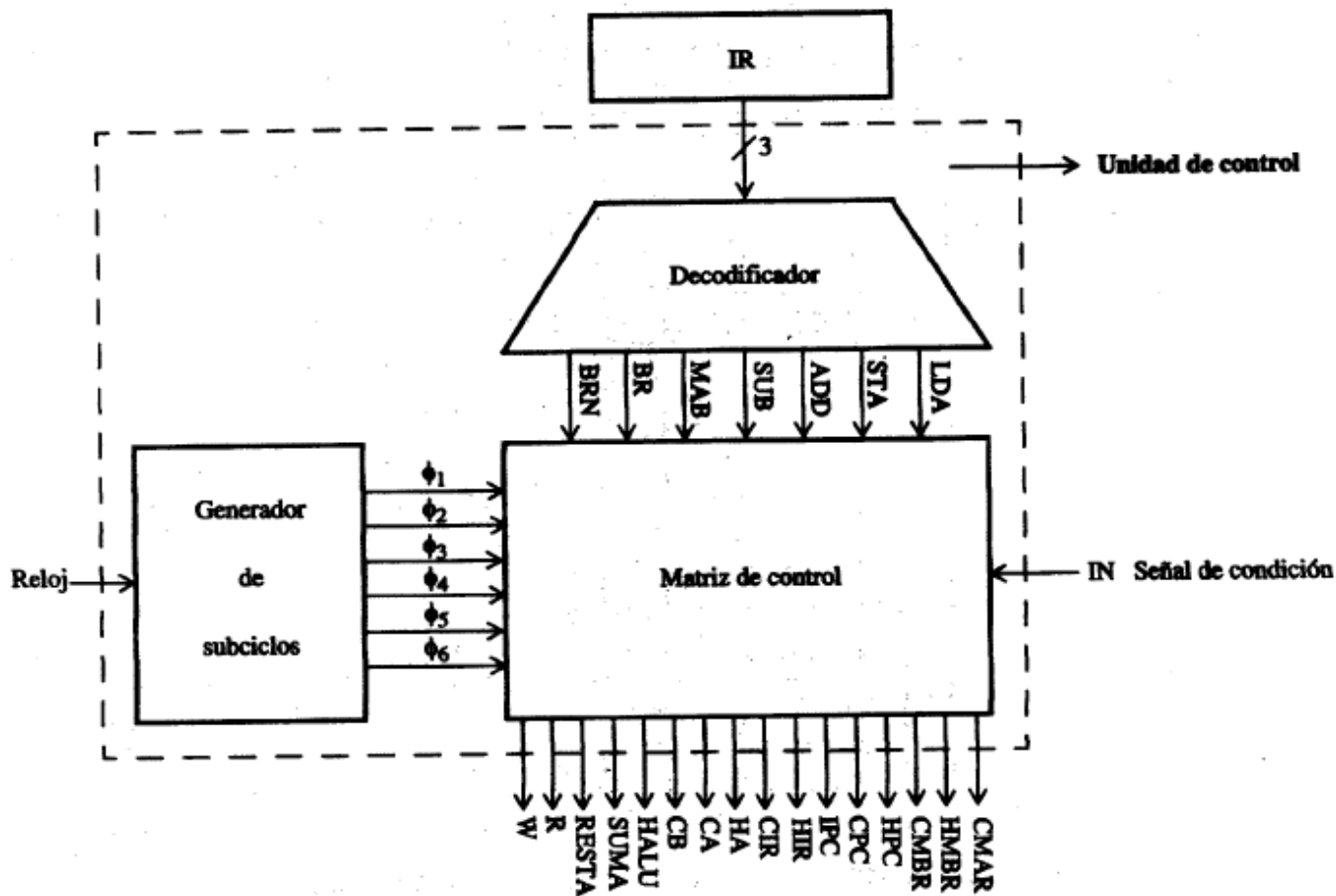
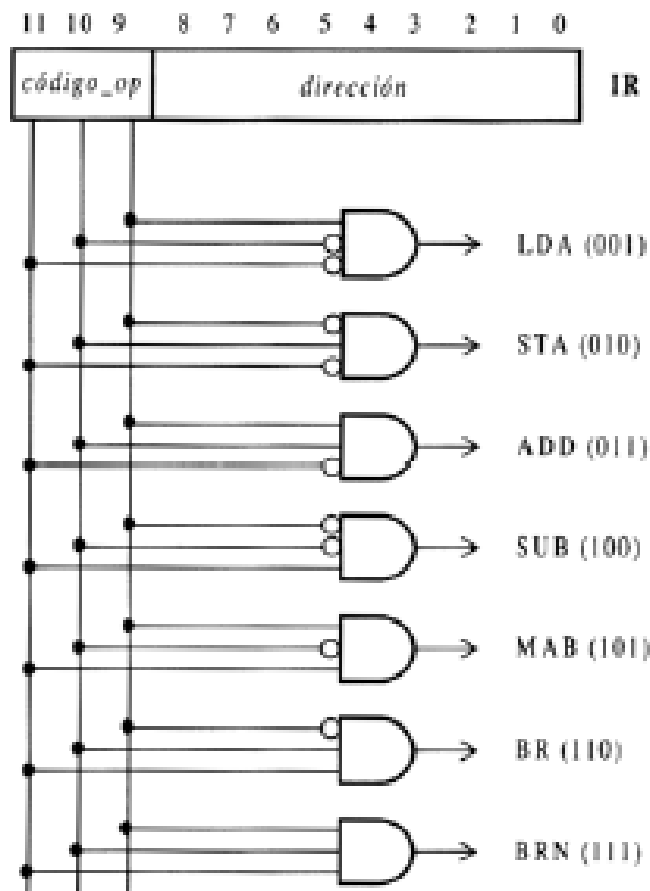


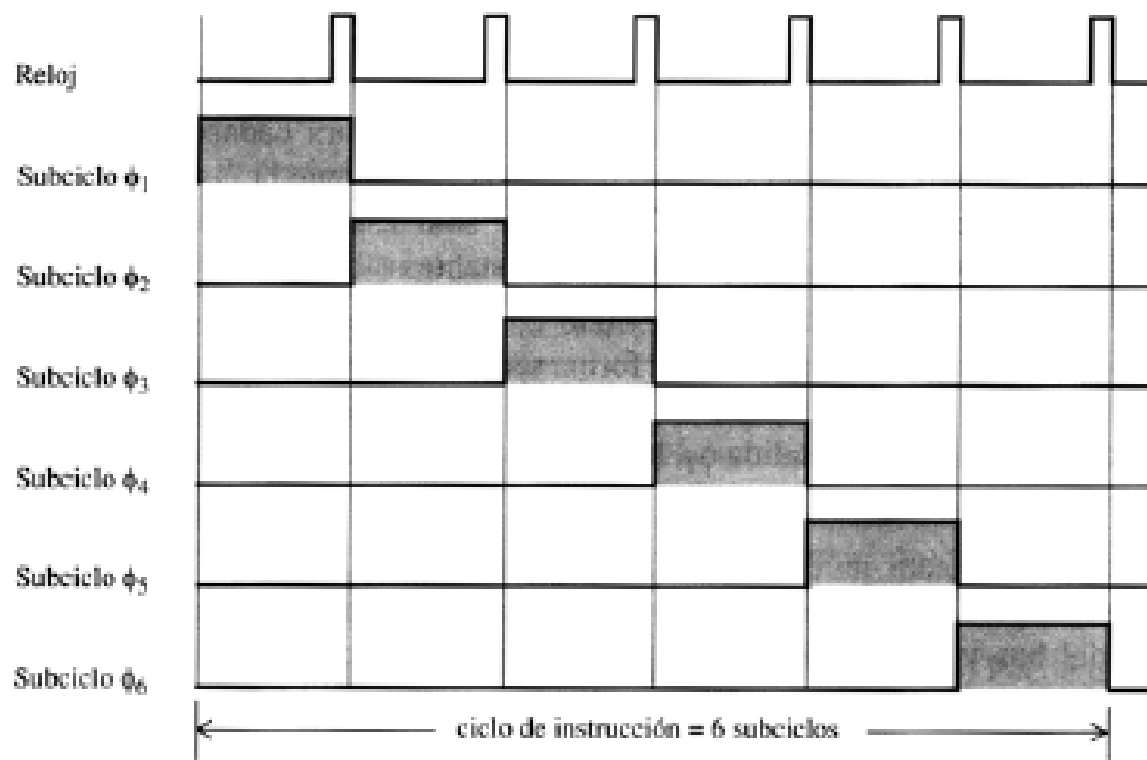
Figura 6.45: Unidad de control del procesador SIMPLE1 con decodificación de sus entradas

# Decodificador del SIMPLE1



Decodificador de instrucciones del procesador SIMPLE1

# División del ciclo de instrucción



División del ciclo de instrucción del procesador SIMPLE1 en 6 subciclos

# Contador en anillo en módulo 6

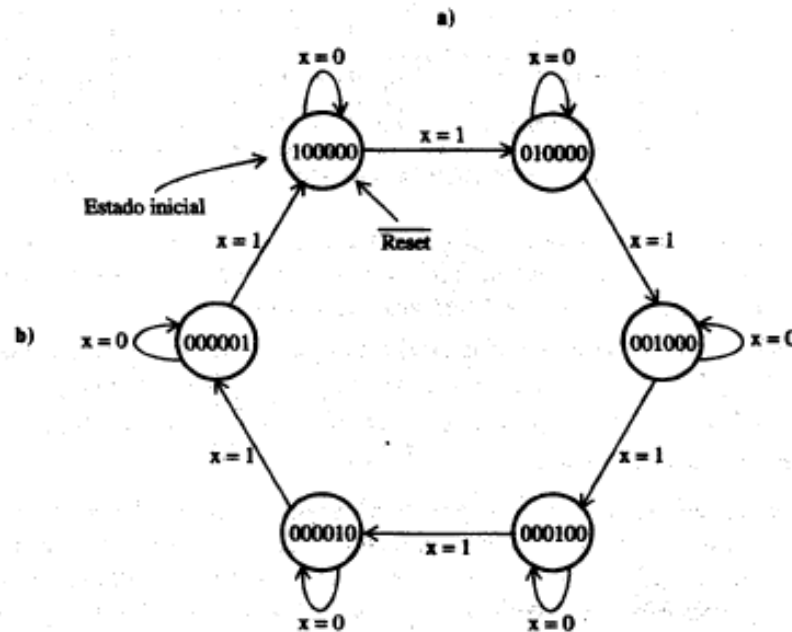
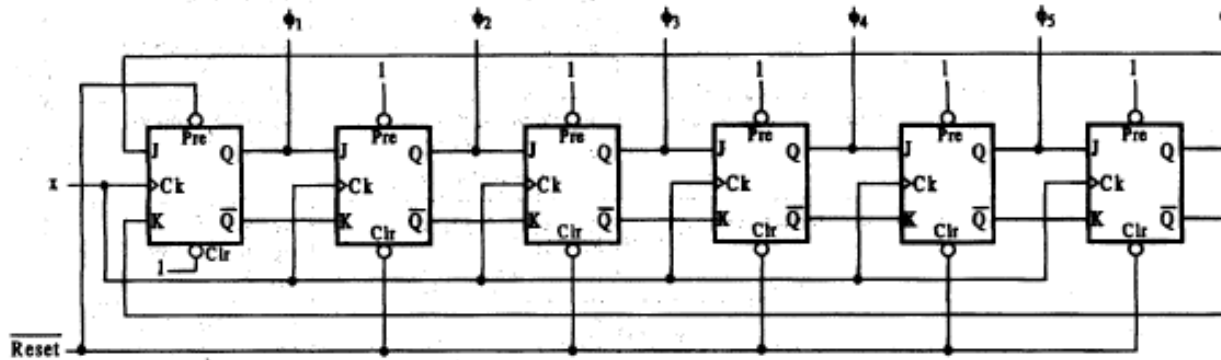


Figura 6.48: Contador en anillo módulo-6: a) Circuito lógico; b) Diagrama de estado

# Señales de control

Activar para cada una de las microoperaciones

	Acción	Microoperaciones	Señales de control
Búsqueda	Fase de búsqueda de la instrucción	$\phi_1$ : MAR $\leftarrow$ PC; $\phi_2$ : MBR $\leftarrow$ M[MAR]; $\phi_3$ : IR $\leftarrow$ MBR, PC $\leftarrow$ PC + 1;	$\phi_1$ : HPC, CMAR $\phi_2$ : R $\phi_3$ : HMBR, CIR, IPC
LDA x	Carga directa	$\phi_4$ : MAR $\leftarrow$ IR(dir); $\phi_5$ : MBR $\leftarrow$ M[MAR]; $\phi_6$ : A $\leftarrow$ MBR;	$\phi_4$ : HIR, CMAR $\phi_5$ : R $\phi_6$ : HMBR, CA
STA dir	Almacenamiento directo	$\phi_4$ : MAR $\leftarrow$ IR(dir); $\phi_5$ : MBR $\leftarrow$ A; $\phi_6$ : M[MAR] $\leftarrow$ MBR;	$\phi_4$ : HIR, CMAR $\phi_5$ : HA, CMBR $\phi_6$ : W
ADD	Suma B a A	$\phi_4$ : A $\leftarrow$ A + B;	$\phi_4$ : SUMA, HALU, CA
SUB	Resta B de A	$\phi_4$ : A $\leftarrow$ A - B;	$\phi_4$ : RESTA, HALU, CA
MAB	Mueve A a B	$\phi_4$ : B $\leftarrow$ A;	$\phi_4$ : HA, CB
BR x	Salto incondicional a x	$\phi_4$ : PC $\leftarrow$ IR(dir);	$\phi_4$ : HIR, CPC
BRN x	Salto a x si indicador negativo a 1	$\phi_4$ : PC $\leftarrow$ IR(dir) (si IN = 1);	$\phi_4$ : si IN = 1: HIR, CPC

Tabla 6.8: Señales de control que hay que activar en cada microoperación

# Matriz l3gica

	IPC	CPC	HPC	CMAR	R	W	CMBR	HMBR	CIR	HIR	CA	HA	SUMA	RESTA	HALU	CB
B3squeda	$\phi_3$		$\phi_1$	$\phi_1$	$\phi_2$			$\phi_3$	$\phi_3$							
LDA				$\phi_4$	$\phi_5$			$\phi_6$		$\phi_4$	$\phi_6$					
STA				$\phi_4$		$\phi_6$	$\phi_5$			$\phi_4$		$\phi_5$				
ADD											$\phi_4$		$\phi_4$		$\phi_4$	
SUB											$\phi_4$			$\phi_4$	$\phi_4$	
MAB												$\phi_4$				$\phi_4$
BR		$\phi_4$								$\phi_4$						
BRN		$\phi_4$ IN								$\phi_4$ IN						

**Tabla 6.9:** Matriz de instantes de activaci3n de las se1ales de control para cada instrucci3n de SIMPLE1

$$IPC = \phi_3$$

$$CPC = \phi_4 BR + \phi_4 IN BRN$$

$$HPC = \phi_1$$

$$CMAR = \phi_1 + \phi_4 LDA + \phi_4 STA$$

$$R = \phi_2 + \phi_5 LDA$$

$$W = \phi_6 STA$$

$$CMBR = \phi_5 STA$$

$$HMBR = \phi_3 + \phi_6 LDA$$

$$CIR = \phi_3$$

$$HIR = \phi_4 LDA + \phi_4 STA + \phi_4 BR + \phi_4 IN BRN$$

$$CA = \phi_6 LDA + \phi_4 ADD + \phi_4 SUB$$

$$HA = \phi_5 STA + \phi_4 MAB$$

$$SUMA = \phi_4 ADD$$

$$RESTA = \phi_4 SUB$$

$$HALU = \phi_4 ADD + \phi_4 SUB$$

$$CB = \phi_4 MAB$$

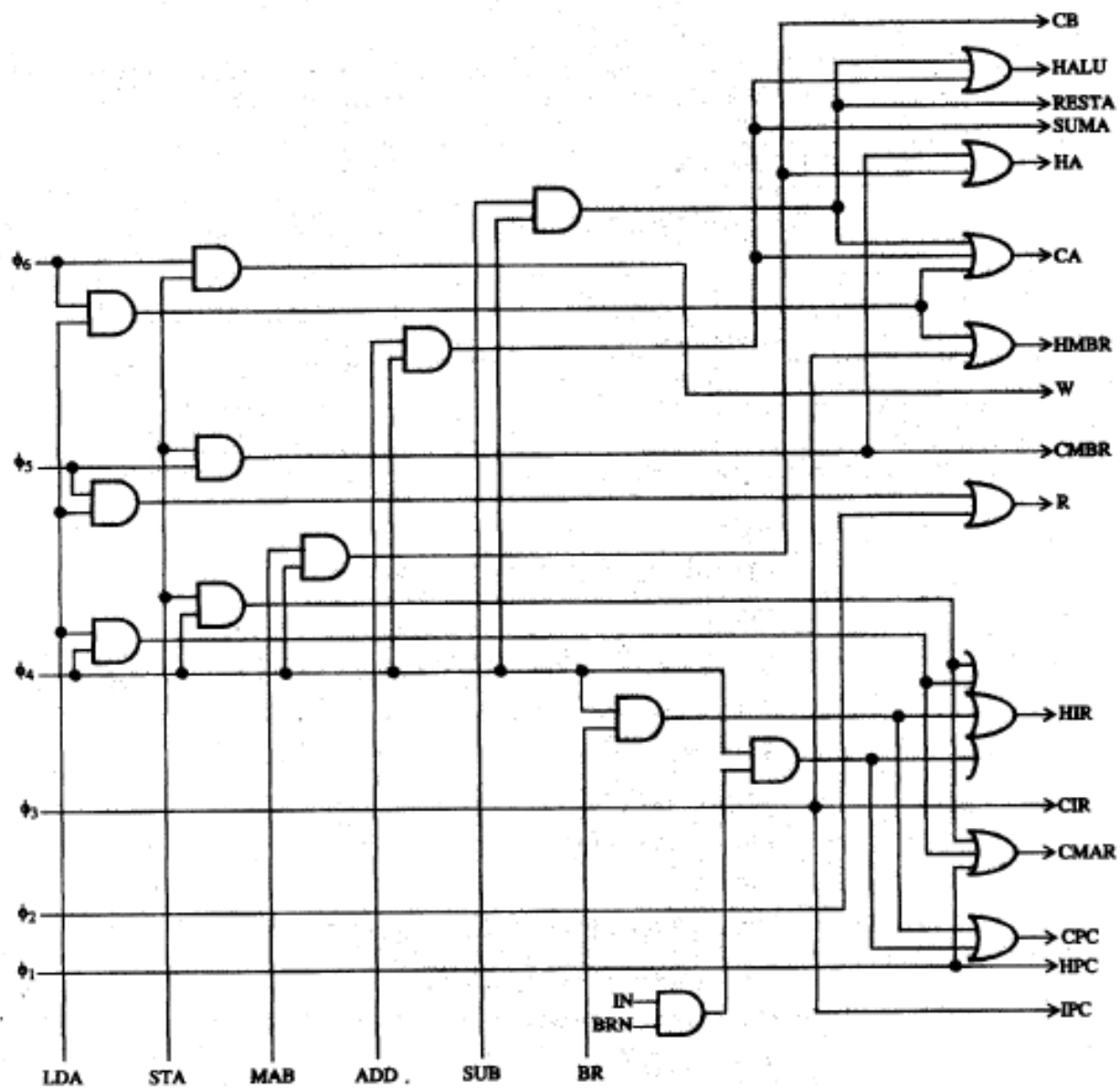


Figura 6.49: Matriz de control del procesador SIMPLE1



## 6.5.7 Diseño de la unidad de procesamiento o ruta de datos

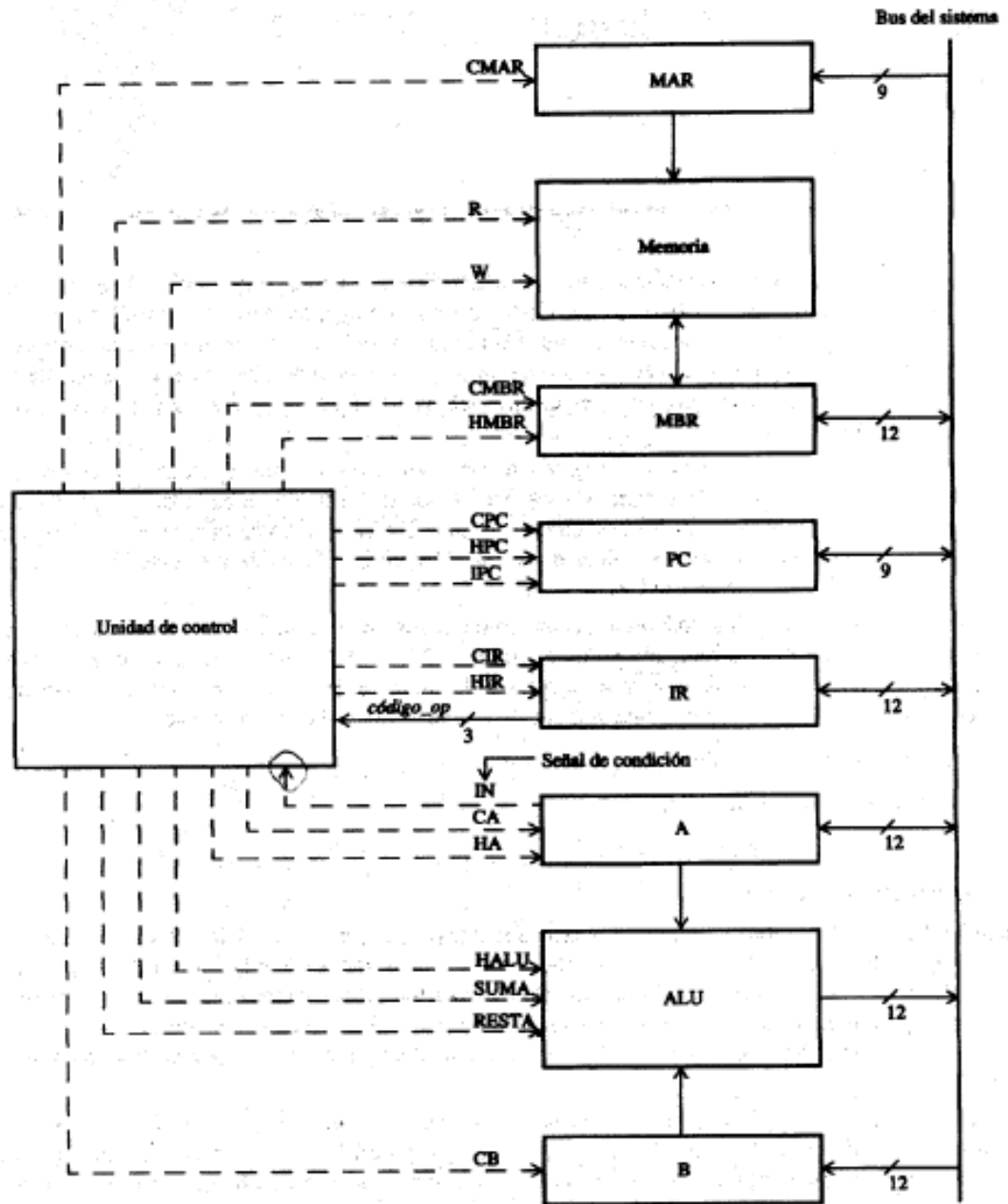


Figura 6.50: Procesador SIMPLE1 y su conexión con la memoria

- En un procesador con instrucciones de cero direcciones (procesador con pila), indique si las secuencias de instrucciones propuestas calculan la expresión  $X=Y^2(X+Z)$ 
  - I. Push[X]; Push[Z]; Add; Push[Y]; Push[Y]; Mult; Mult; Pop[X]
  - II. Push[Y]; Push[Y]; Push[X]; Push[Z]; Add; Mult; Mult; Pop[X]
- A) I:sí, II:sí
- B) I:sí, II:no
- C) I:no, II:sí
- D) I:no, II:no

- En un procesador con instrucciones de cero direcciones, indicar cuantos accesos a memoria se necesitan para completar la secuencia  $Y = (X+Y)+Z$ .
  - A) 10.
  - B) 6.
  - C) 12.
  - D) Ninguna de las anteriores.

- Solución

- [Ver el problema 6-4 y el apartado 6.1.4 del texto de teoría.]
- En el caso de un procesador de cero direcciones el cálculo de la expresión dada se puede realizar mediante el conjunto de 6 instrucciones siguientes: Push[X]; Push[Y]; Add; Push[Z]; Add; Pop[Y]
- Serán necesarios 6 accesos a memoria para leer el código de operación de cada instrucción;
- además en el caso de las instrucciones Push y Pop se necesita un acceso adicional a memoria para acceder a los operandos, ya que las operaciones aritméticas se realizan con operandos que se encuentran ya en la pila.
- En total son 6 instrucciones, tres de las cuales son Push y una Pop:  $6+3+1 = 10$  accesos
- Respuesta: A

- Empleado un procesador de una dirección (procesador con acumulador) con un banco de registro  $R_i$ , indicar qué operación calcula la secuencia de instrucciones:
  - Load X; Add Y; Add Z; Mult X, Store R1, Mult R1, Div X, Store X.
    - A)  $X = ((X+Y+Z)^2)X^2$
    - B)  $X = ((X+Y+Z)^2)X$
    - C)  $X = (X+Y+Z)^2$
    - D) Ninguna de las anteriores
- El resultado de la ejecución de la secuencia de instrucciones propuesta es:
- $$\left( \left( (X + Y + Z) X \right)^2 \right) / X = \left( (X + Y + Z)^2 X^2 \right) / X = (X + Y + Z)^2 X$$
- Respuesta: B=  $( X = (( X + Y + Z )^2 ) X )$