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

6.6 Microprogramación

6.1 Modelo original de Wilkes

6.2 Estructura de una unidad de control microprogramada

6.2.1 Conceptos básicos

6.2.2 Elementos de una unidad de control microprogramada

6.2.3 Secuenciamiento de las microinstrucciones

6.2.4 Organización de la memoria de control

6.2.5 Ejecución de las microinstrucciones

DISEÑO DEL PROCESADOR

- **Partes de un sistema digital**
  - Unidad de procesamiento:
    - Se almacenan y transforman los datos
  - Unidad de control:
    - Genera las secuencias se señales de control de acuerdo al algoritmo de transferencia de registros.

Realización cableada

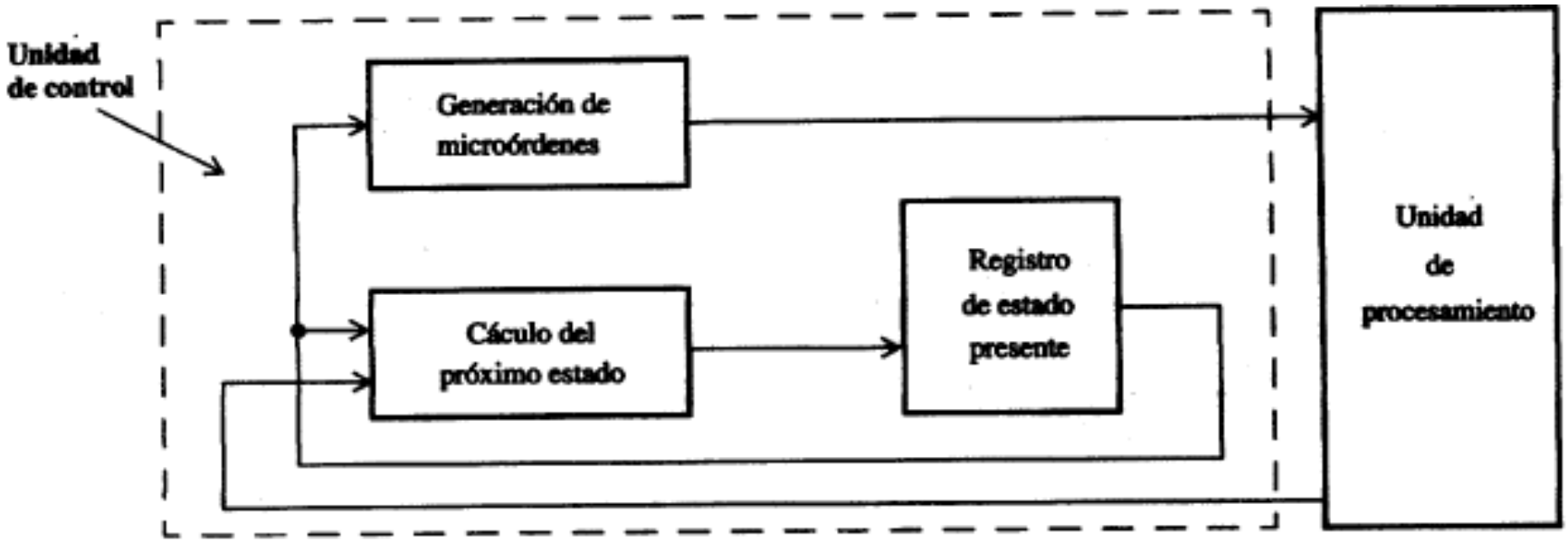


Figura 6.2: Estructura de una unidad de control con lógica cableada

Realización microprogramada

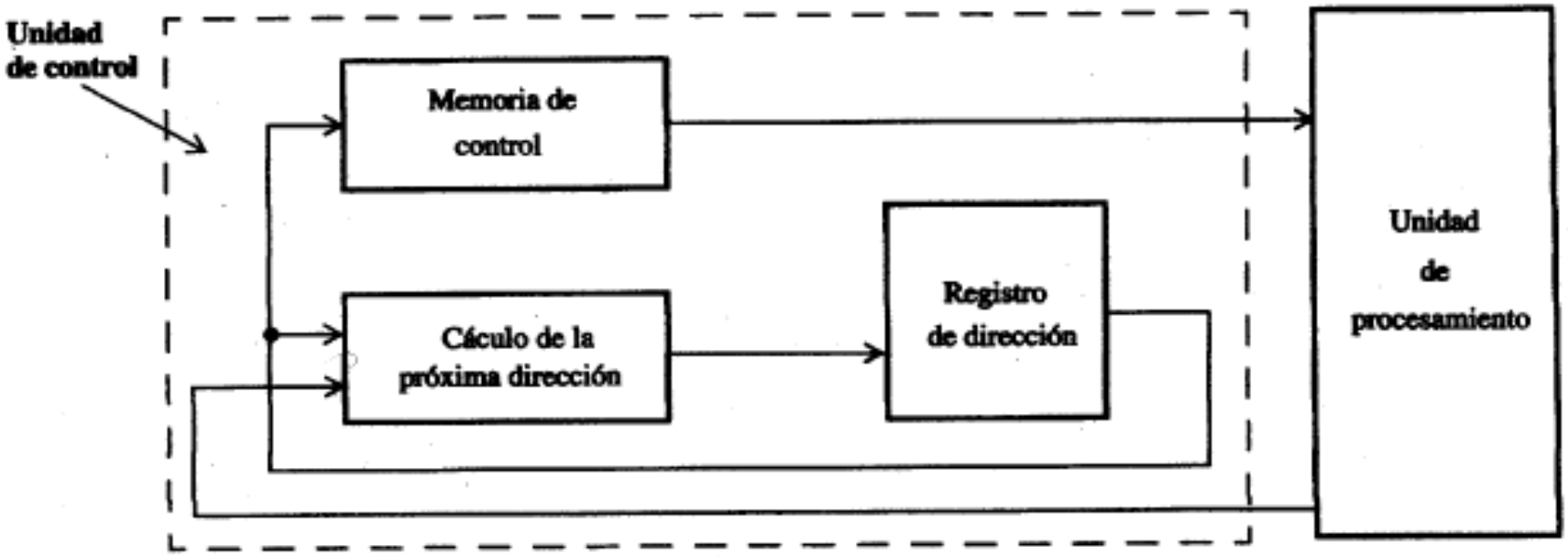


Figura 6.3: Estructura de una unidad de control microprogramada

Tipos de microoperaciones

- De transferencia
- De proceso

Fases en el ciclo de ejecución de una instrucción

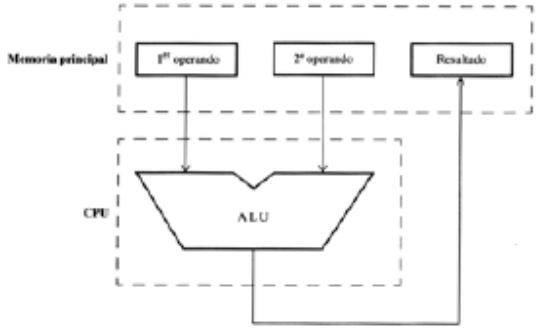
- Búsqueda
- Decodificación
- Búsqueda de los operandos
- Ejecución

6.1 Repertorio de instrucciones

- Cada CPU tiene su propio y especifico formato de instrucciones
- Una instrucción es una cadena de bits que se agrupan en campos con tamaños diferentes
  - Las instrucciones pueden tener , número de campos distintos, de longitudes diferentes => instrucciones de longitudes 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

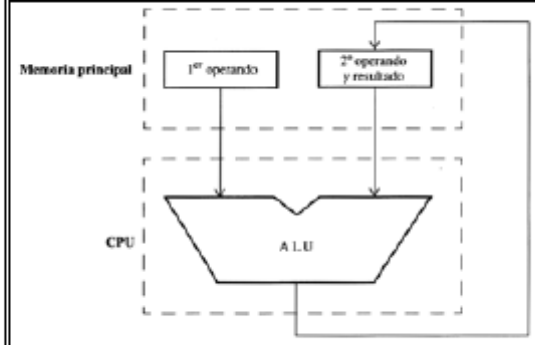
6.1.1 Procesadores de tres direcciones

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

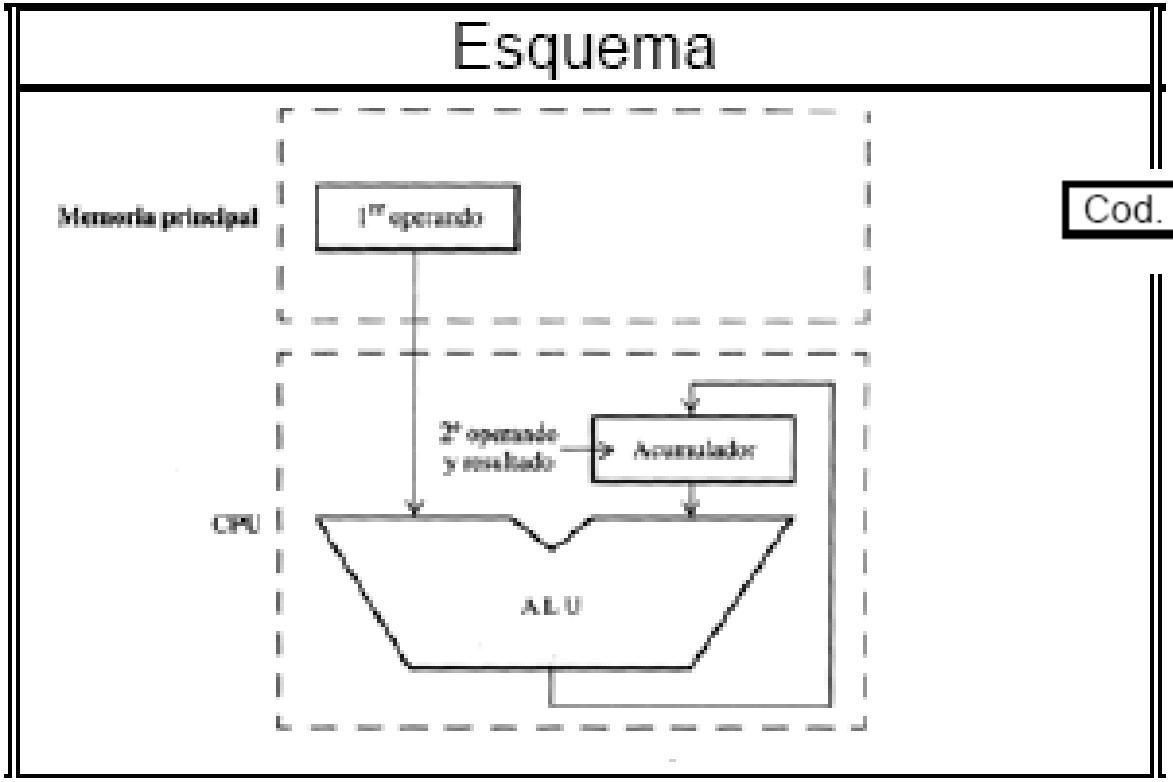
	Esquema	Formato de la instrucción
Procesadores de 3 direcciones		Cod. Opera. | Operan. 1 | Operan. 2 | Resultado

6.1.2 Procesadores de dos direcciones

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

	Esquema	Formato de la instrucción
Procesadores de 2 direcciones		

6.1.3 Procesadores de una dirección (procesadores con acumulador)

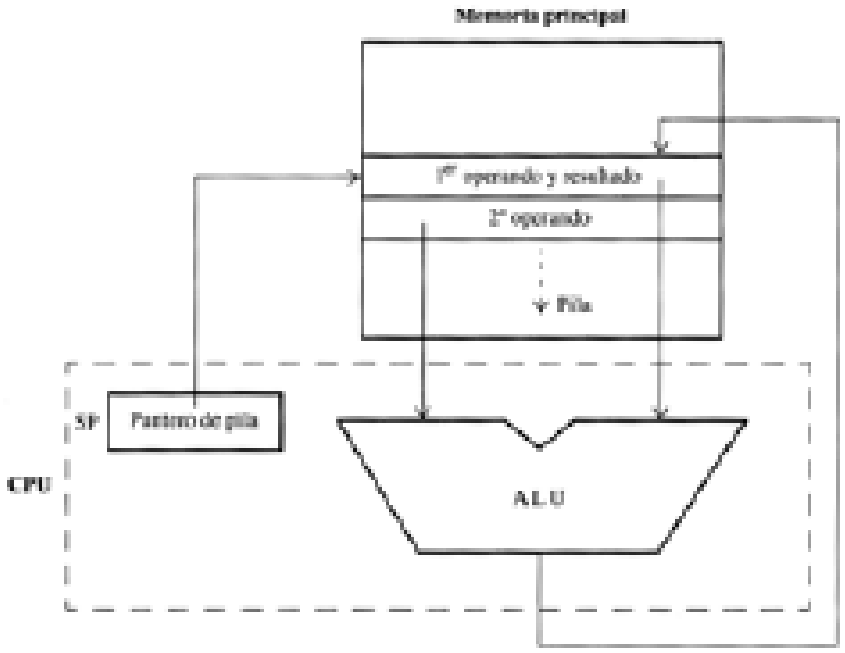


- El procesador dedica un registro como operando destino

6.1.4 Procesadores de cero direcciones (procesadores con pila)

- Primer, segundo operando y resultado en la pila

Esquema

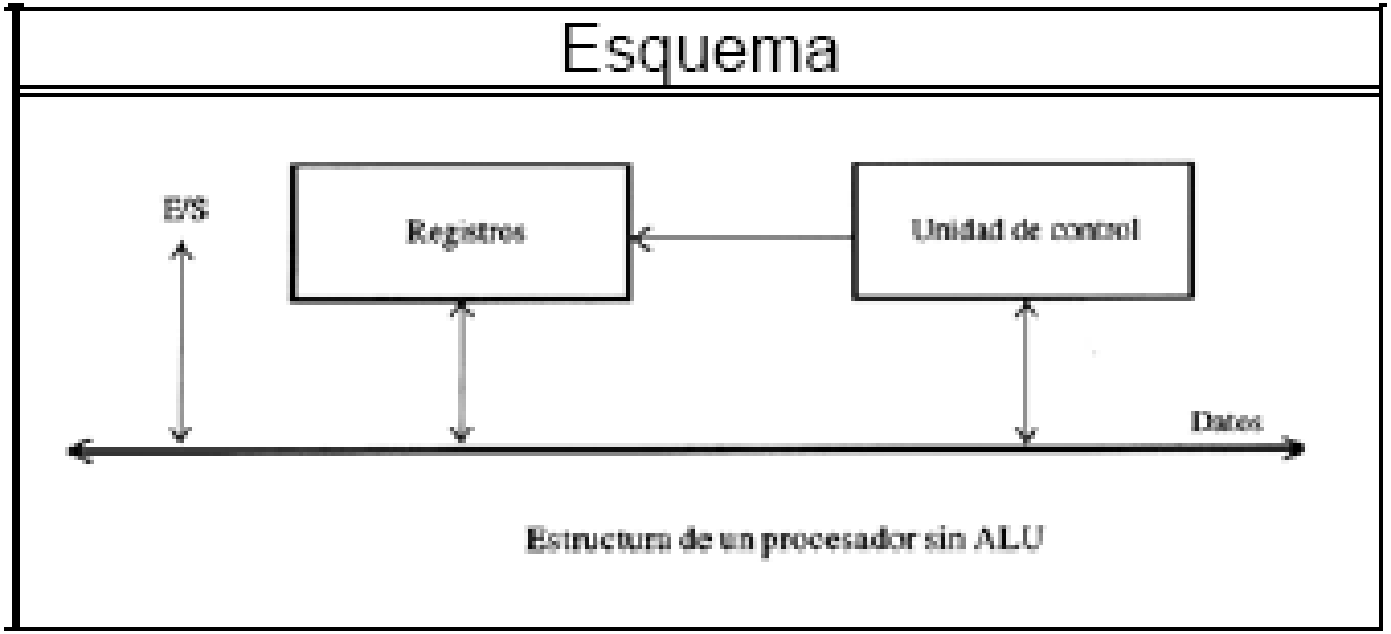


Flujo de datos en un procesador de pila en las instrucciones con referencia a memoria

Código Operación

6.1.5 Procesadores sin ALU

- Operaciones de transferencia
- Las operaciones aritmético-lógicas debe realizarse mediante operaciones de desplazamiento



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
  - Aumenta el número de accesos a memoria

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)

Procesador RISC	Procesador CISC
1) Los accesos a memoria se restringen a instrucciones de carga/almacenamiento y las de manipulación de datos son entre registros.	1) La mayoría de los tipos de instrucciones permiten que el acceso a memoria sea de forma directa.
2) Existe un número limitado de modos de direccionamiento.	2) Hay un número considerable de modos de direccionamiento.
3) Los formatos de las instrucciones tienen todos la misma longitud.	3) Los formatos de las instrucciones tienen longitudes diferentes.
4) Las instrucciones realizan operaciones elementales.	4) Las instrucciones realizan operaciones elementales y complejas.

Comparación de las características de un procesador RISC frente a un procesador CISC

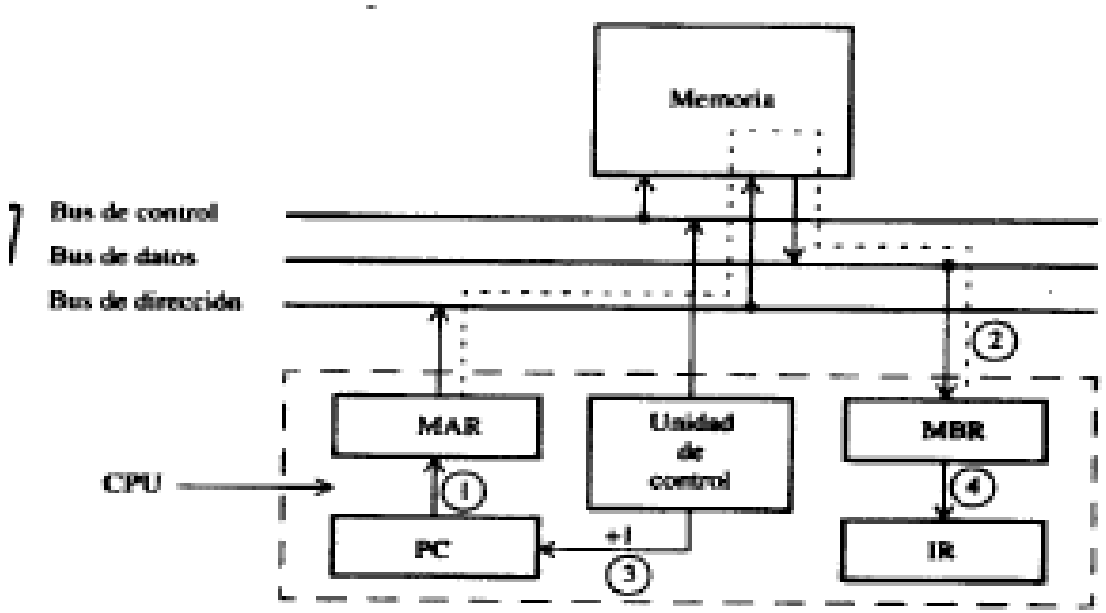
6.2 Modos de direccionamiento

- 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 encuentra 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

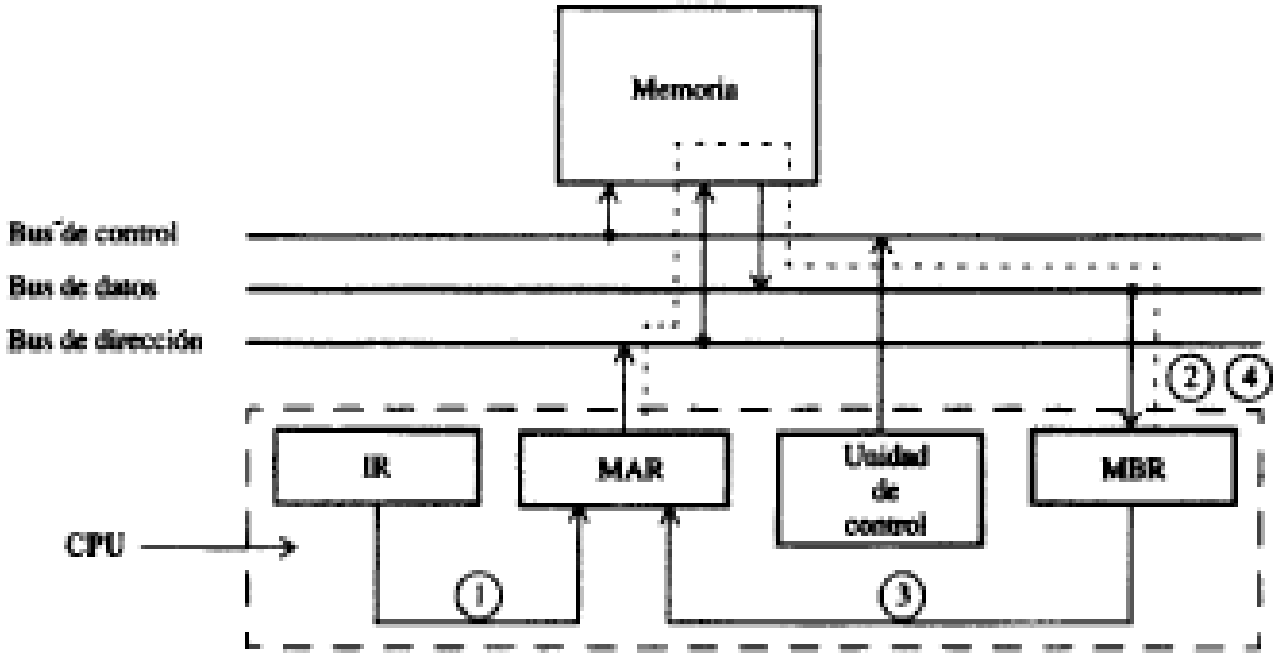
6.3 Ciclo de ejecución de una instrucción

- 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

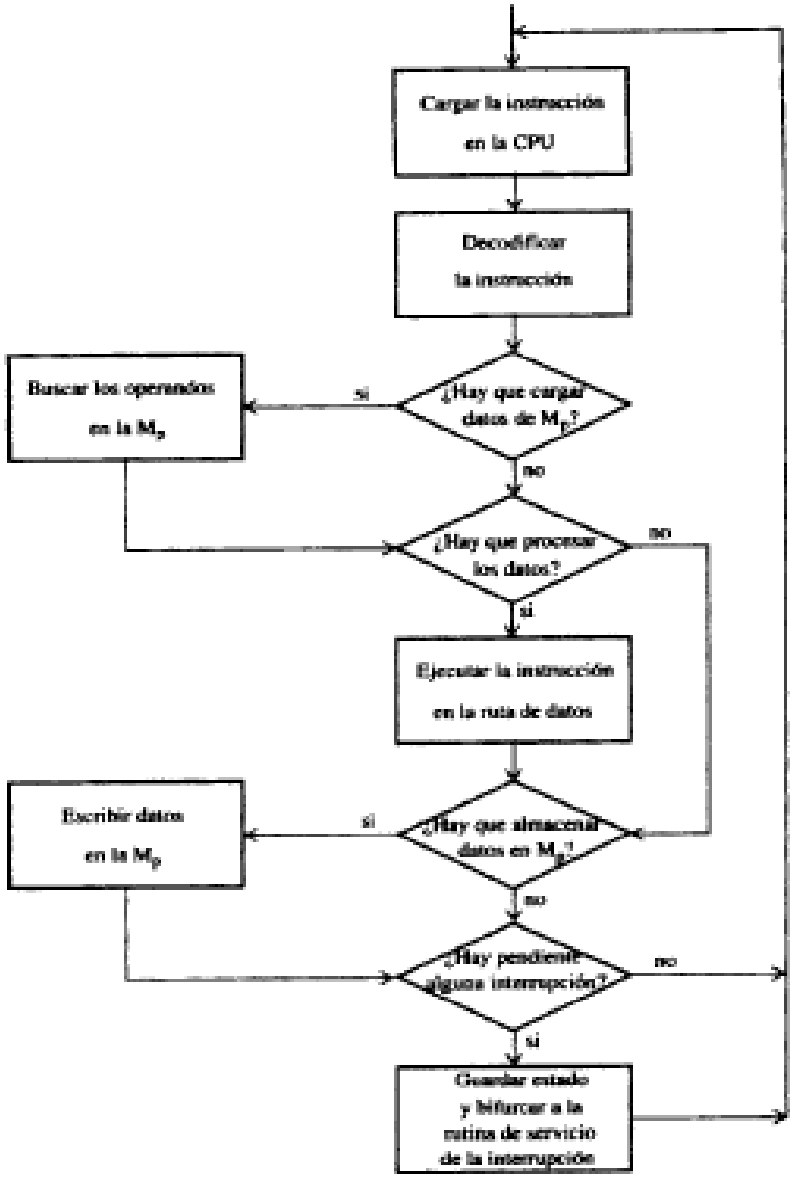
Flujo de datos en la fase de búsqueda de la instrucción



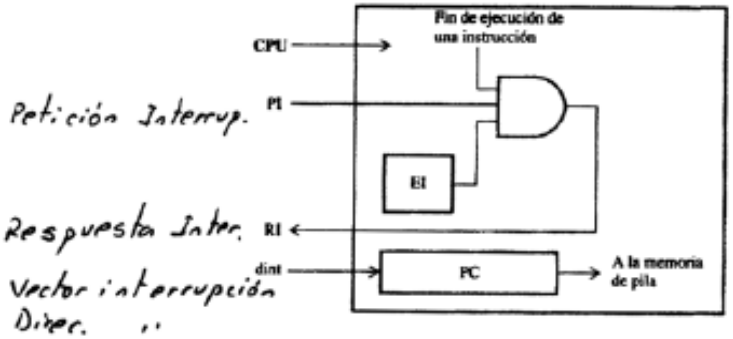
Fase de búsqueda de operandos con direccionamiento indirecto



Flujo de datos en la fase de búsqueda de los operandos con un modo de direccionamiento indirecto

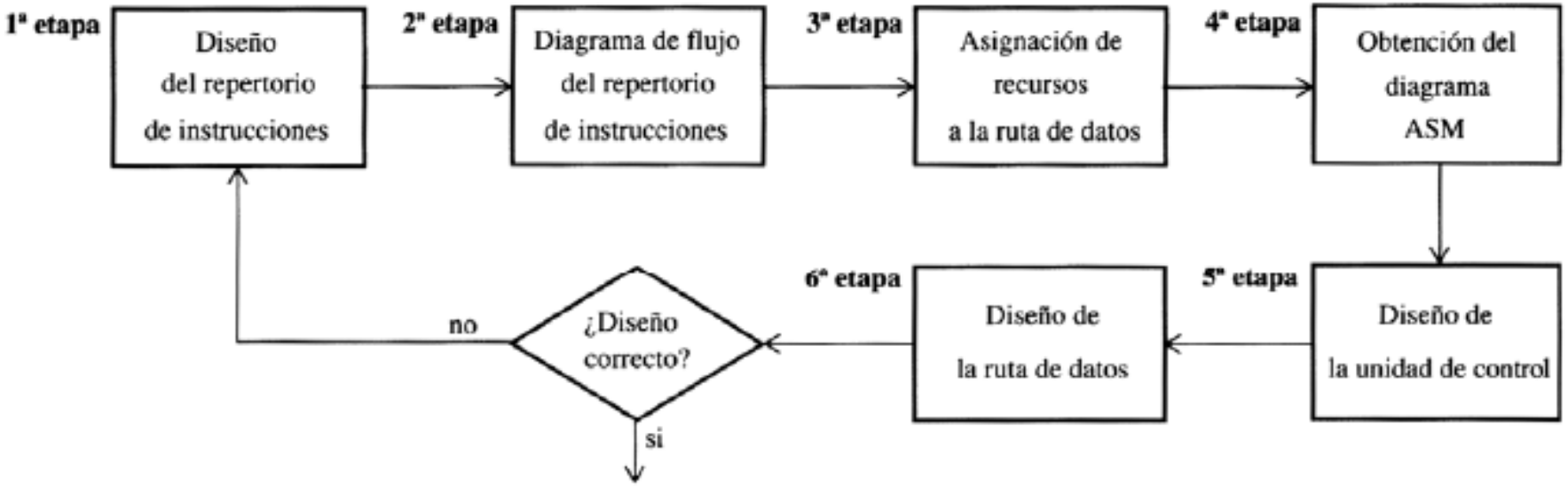


Organigrama del secuenciamiento de ciclos de instrucción y/o ciclos de interrupción



Gestión del ciclo de interrupción por el procesador

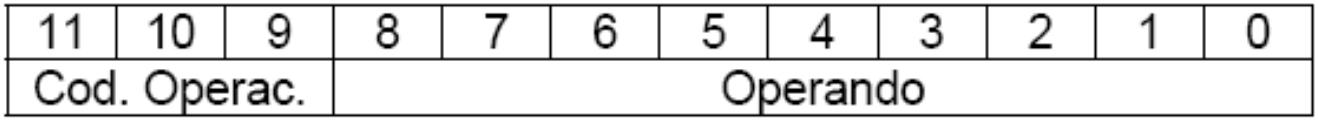
6.4 Fases en el diseño del procesador



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:



6.5.2 Repertorio de instrucciones

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

Repertorio de instrucciones de SIMPLE1

6.5.3 Diagrama de flujo del repertorio de instrucciones

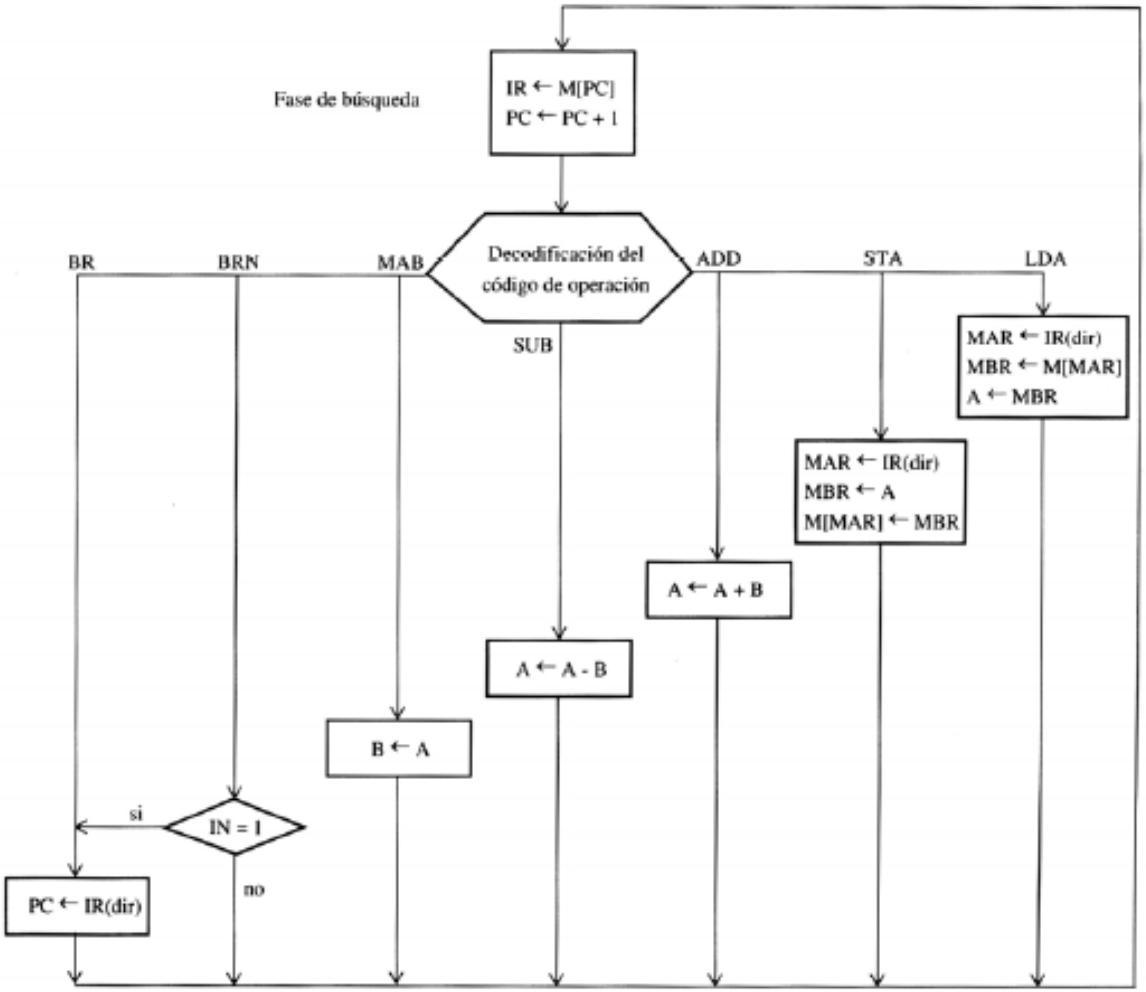
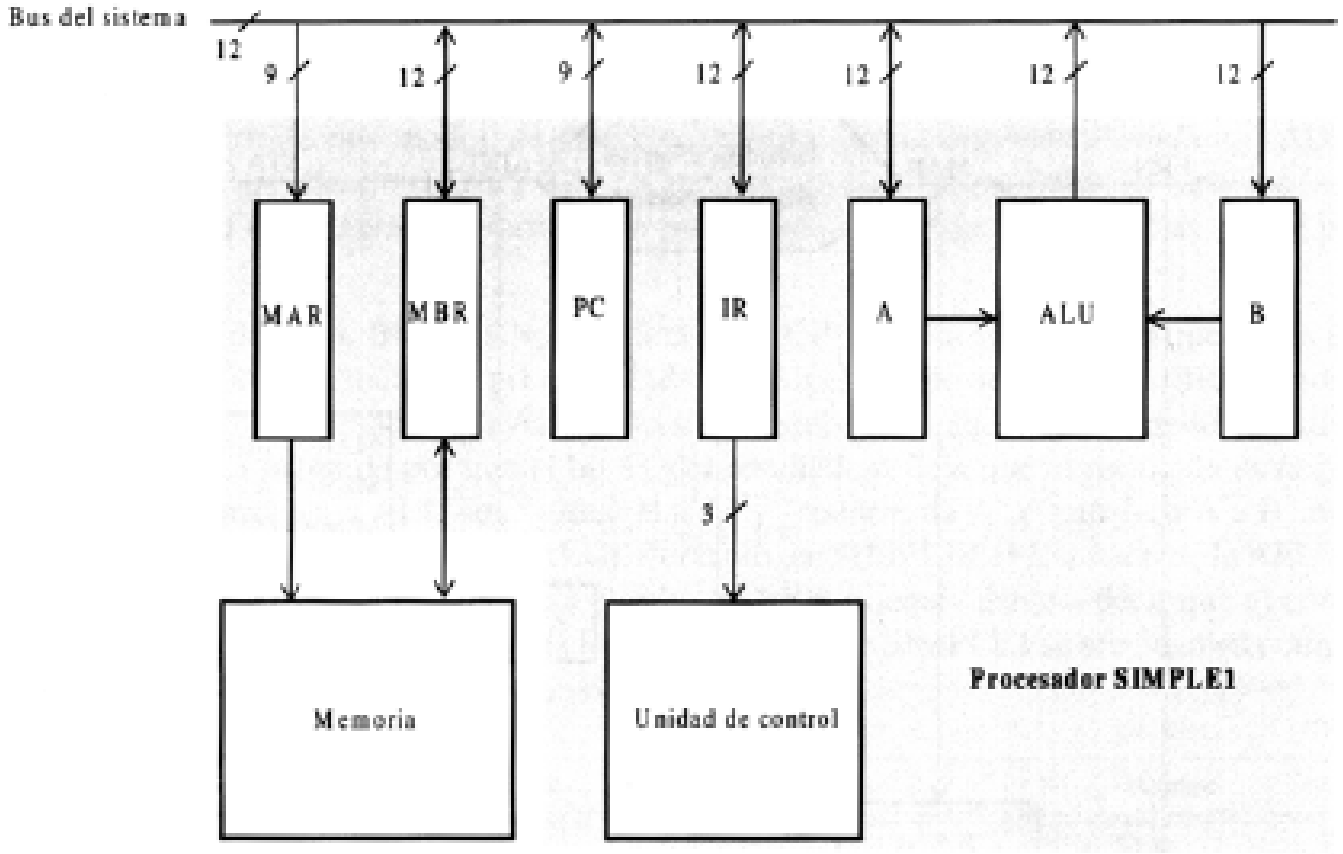


Diagrama de flujo del repertorio de instrucciones de SIMPLE1

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

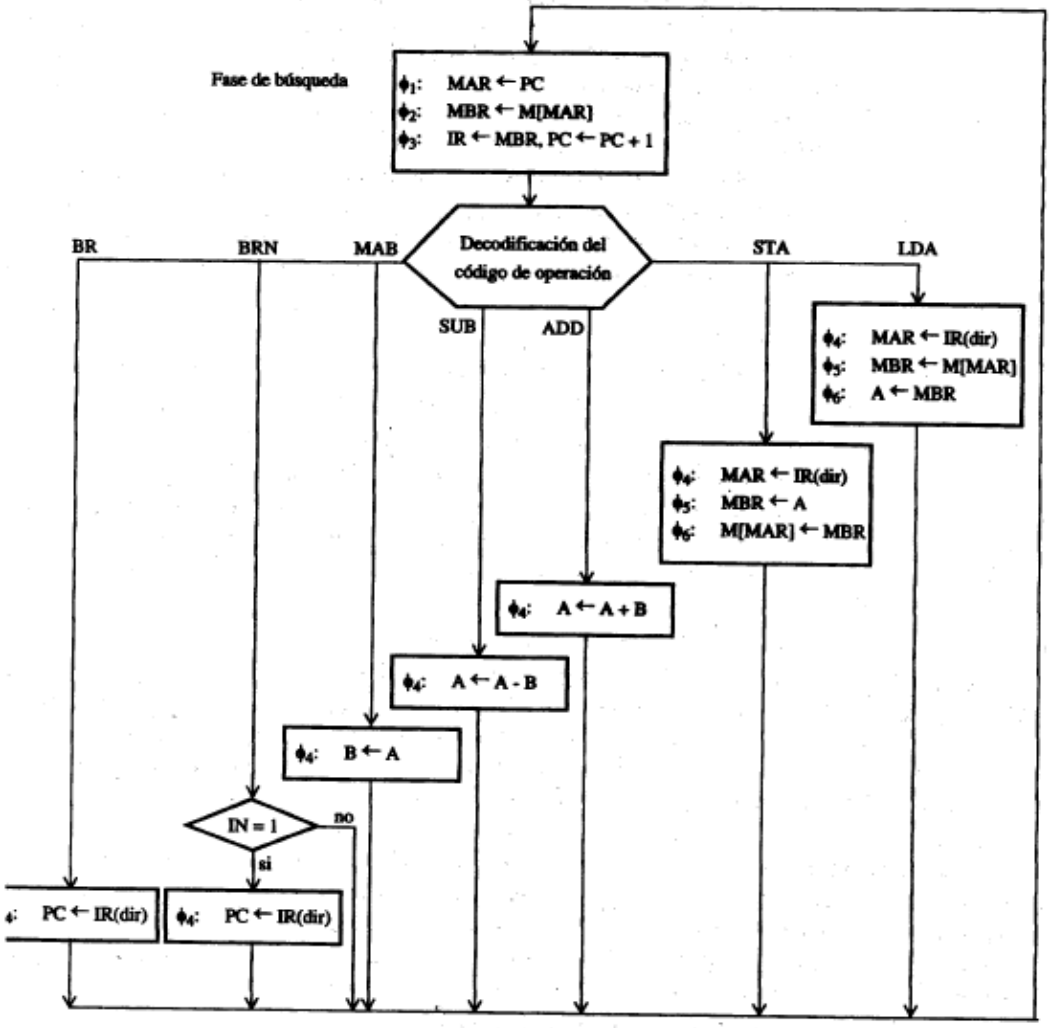
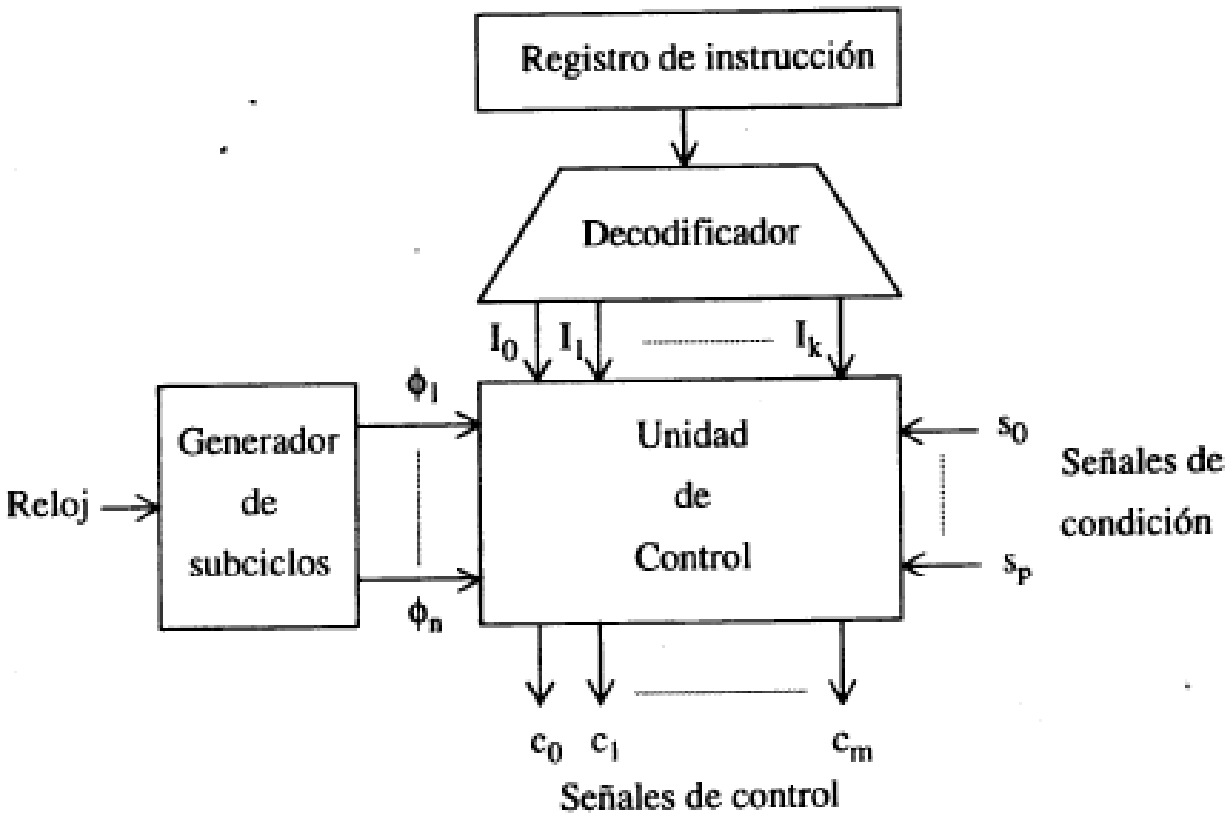


Figura 6.43: Diagrama ASM del procesador SIMPLE1

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 ← M[MAR])
W	Escribir en la memoria (M[MAR] ← MBR)
CMAR	Cargar el contenido del bus en MAR (MAR ← Bus)
HMBR	Habilitar el registro MBR (Bus ← MBR)
CMBR	Cargar el contenido del bus en MBR (MBR ← Bus)
HPC	Habilitar el registro PC (Bus ← PC)
CPC	Cargar el contenido del bus en PC (PC ← Bus)
IPC	Incrementar el contenido de PC (PC ← PC + 1)
HIR	Habilitar el registro IR (Bus ← IR)
CIR	Cargar el contenido del bus en IR (IR ← Bus)
HA	Habilitar el registro A (Bus ← A)
CA	Cargar el contenido del bus en A (A ← Bus)
CB	Cargar el contenido del bus en B (B ← 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

Señales de control del procesador SIMPLE1

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

Unidad de Control del SIMPLE1

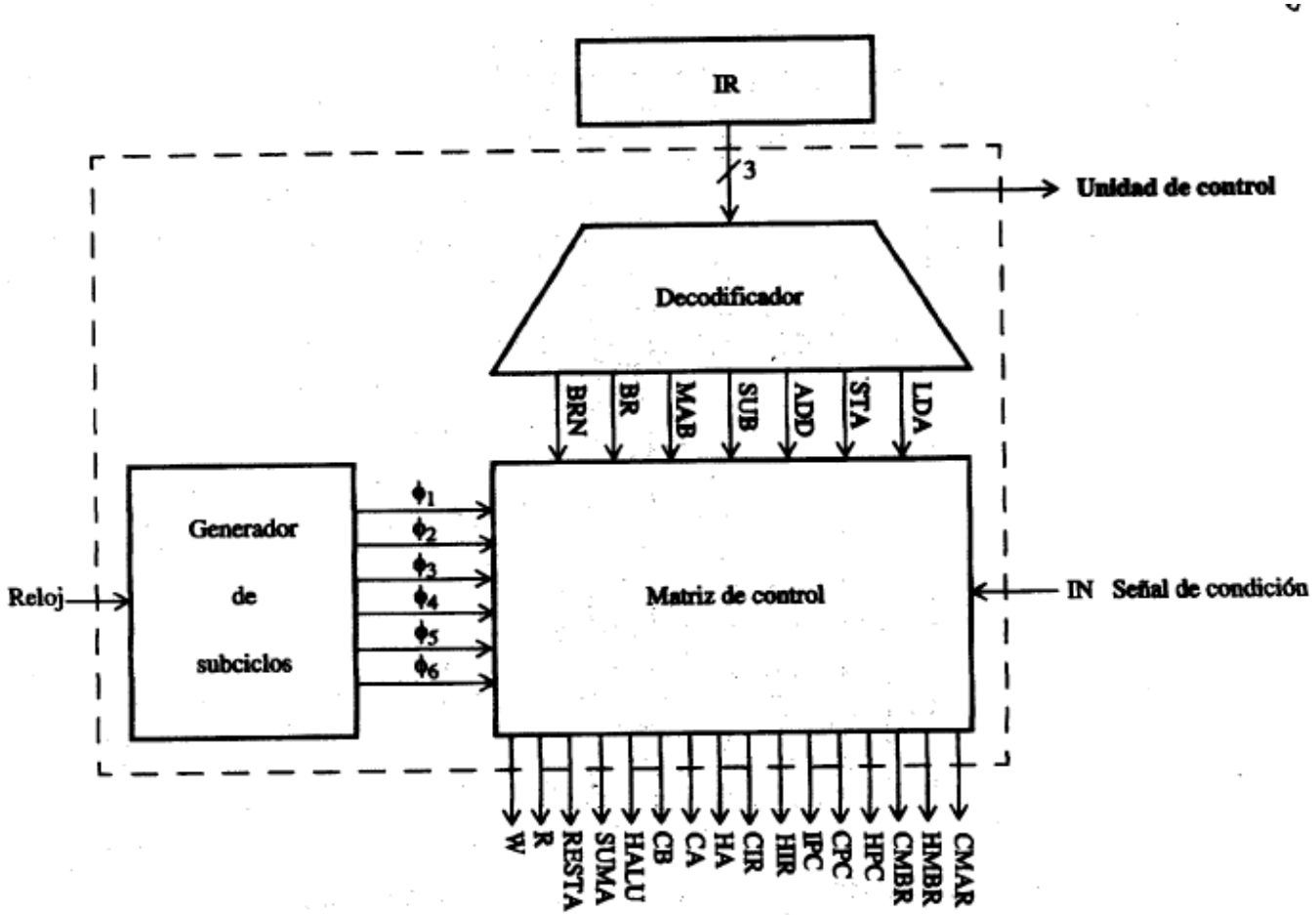
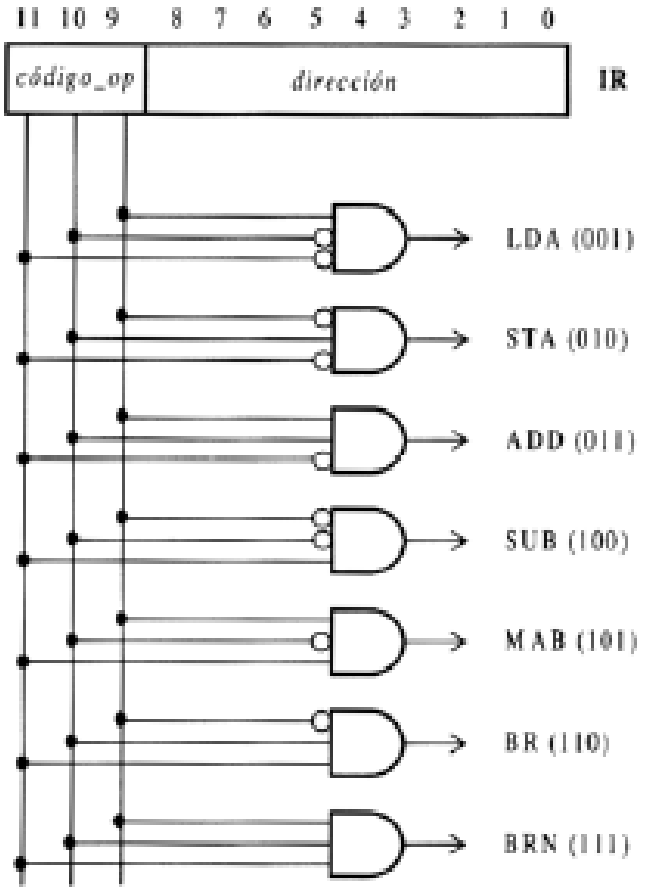


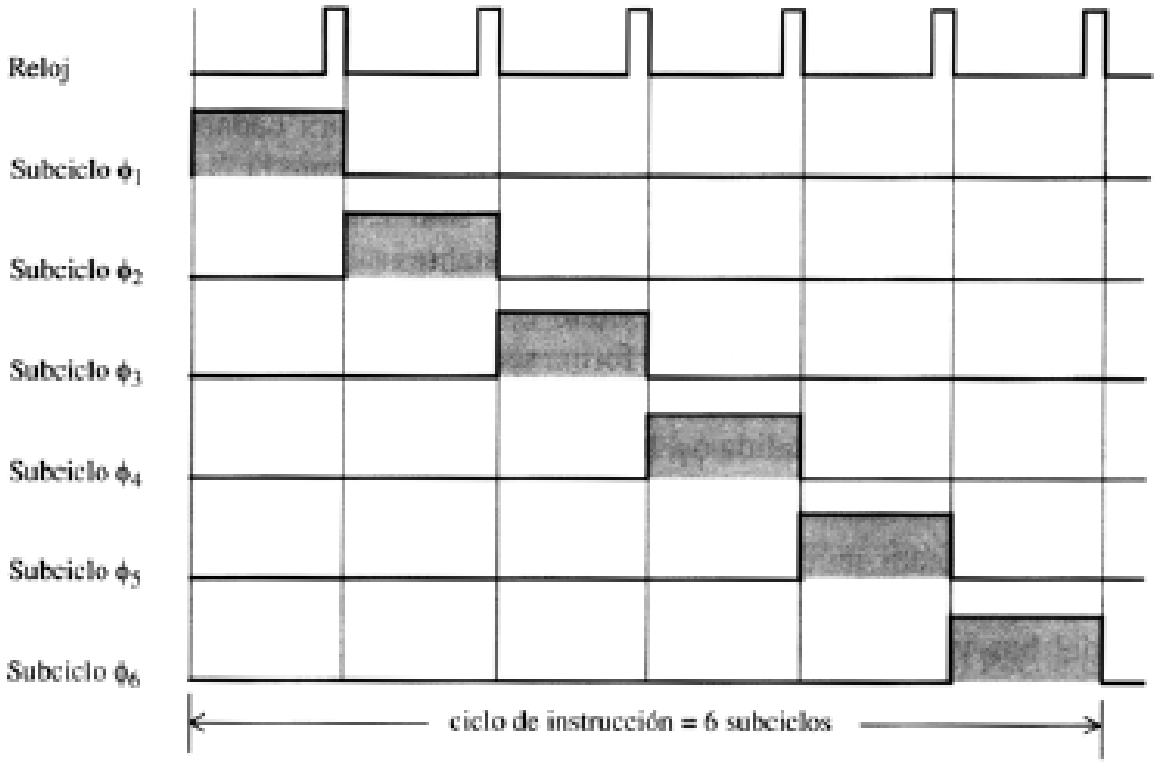
Figura 6.45: Unidad de control del procesador SIMPLE1 con decodificación de sus entradas

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

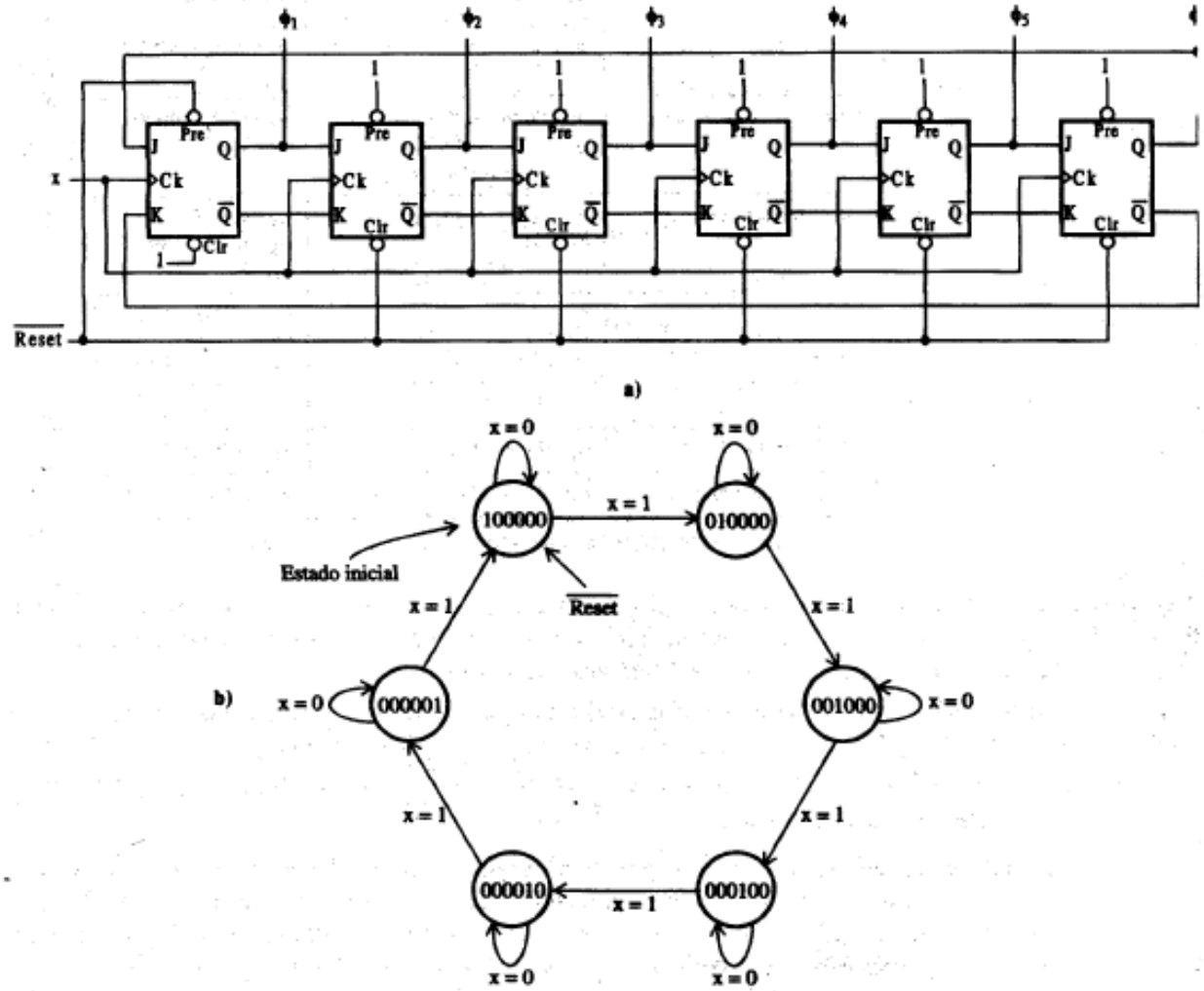


Figura 6.48: Contador en anillo módulo-6: a) Circuito lógico; b) Diagrama de estado

Señales de control

activar para cada una de las microoperaciones.

	Acción	Microoperaciones	Señales de control
Búsqueda	Fase de búsqueda de la instrucción	ϕ_1: MAR ← PC; ϕ_2: MBR ← M[MAR]; ϕ_3: IR ← MBR, PC ← PC + 1;	ϕ_1: HPC, CMAR ϕ_2: R ϕ_3: HMBR, CIR, IPC
LDA x	Carga directa	ϕ_4: MAR ← IR(dir); ϕ_5: MBR ← M[MAR]; ϕ_6: A ← MBR;	ϕ_4: HIR, CMAR ϕ_5: R ϕ_6: HMBR, CA
STA dir	Almacenamiento directo	ϕ_4: MAR ← IR(dir); ϕ_5: MBR ← A; ϕ_6: M[MAR] ← MBR;	ϕ_4: HIR, CMAR ϕ_5: HA, CMBR ϕ_6: W
ADD	Suma B a A	ϕ_4: A ← A + B;	ϕ_4: SUMA, HALU, CA
SUB	Resta B de A	ϕ_4: A ← A - B;	ϕ_4: RESTA, HALU, CA
MAB	Mueve A a B	ϕ_4: B ← A;	ϕ_4: HA, CB
BR x	Salto incondicional a x	ϕ_4: PC ← IR(dir);	ϕ_4: HIR, CPC
BRN x	Salto a x si indicador negativo a 1	ϕ_4: PC ← IR(dir) (si IN = 1);	ϕ_4: si IN = 1: HIR, CPC

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

Matriz lógica

	IPC	CPC	HPC	CMAR	R	W	CMBR	HMBR	CIR	HIR	CA	HA	SUMA	RESTA	HALU	CB
Búsqueda	ϕ_3		ϕ_1	ϕ_1	ϕ_2			ϕ_3	ϕ_3							
LDA				ϕ_4	ϕ_5			ϕ_6		ϕ_4	ϕ_6					
STA				ϕ_4		ϕ_6	ϕ_5			ϕ_4		ϕ_5				
ADD											ϕ_4		ϕ_4		ϕ_4	
SUB											ϕ_4			ϕ_4	ϕ_4	
MAB												ϕ_4				ϕ_4
BR		ϕ_4								ϕ_4						
BRN		ϕ_4 IN								ϕ_4 IN						

Tabla 6.9: Matriz de instantes de activación de las señales de control para cada instrucción de SIMPLE1

$$IPC = \phi_3$$

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

$$HPC = \phi_1$$

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

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

$$W = \phi_6 STA$$

$$CMBR = \phi_5 STA$$

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

$$CIR = \phi_3$$

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

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

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

$$SUMA = \phi_4 ADD$$

$$RESTA = \phi_4 SUB$$

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

$$CB = \phi_4 MAB$$

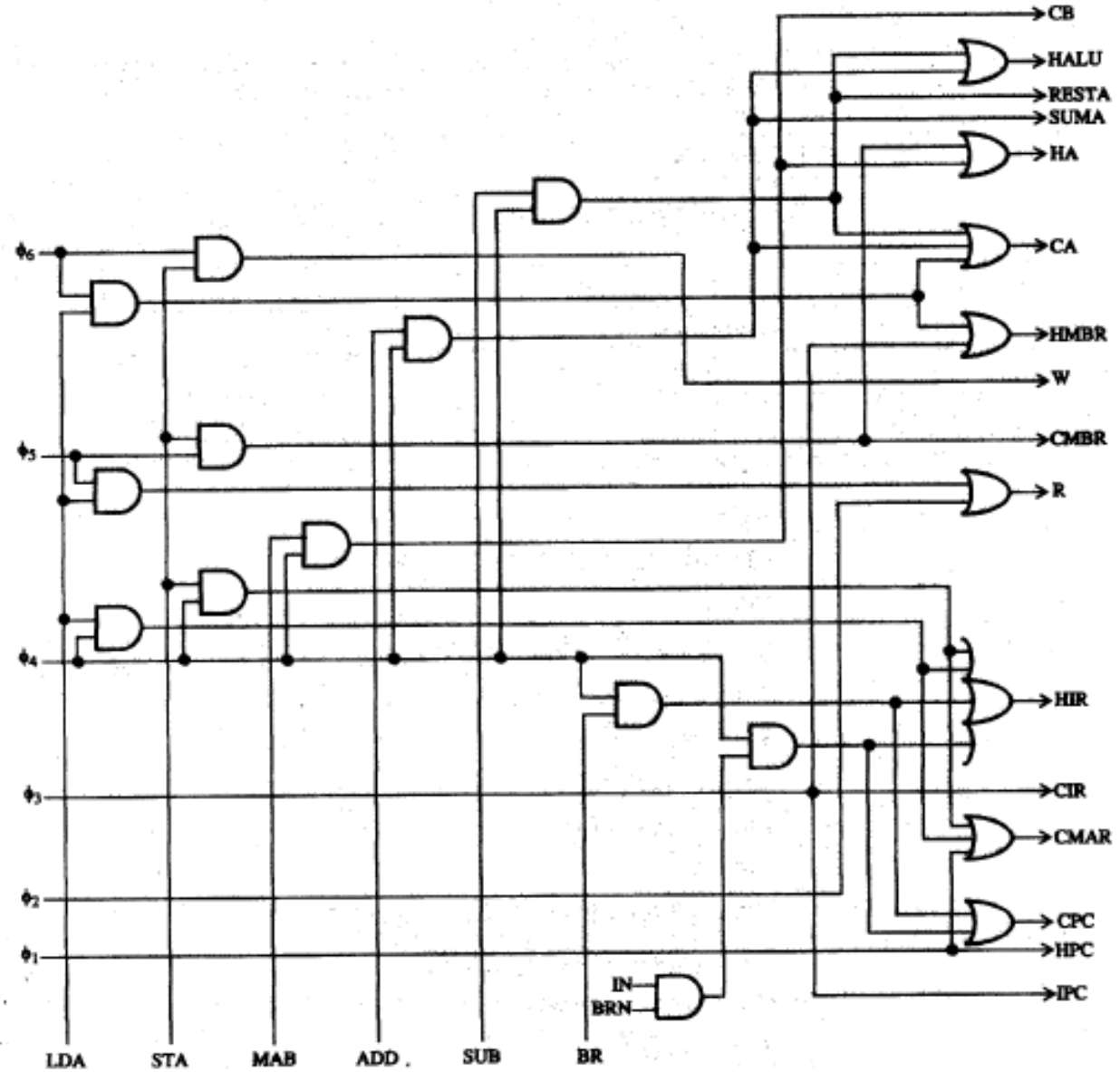


Figura 6.49: Matriz de control del procesador SIMPLE1

6.5.7 Diseño de la unidad de procesamiento o ruta de datos

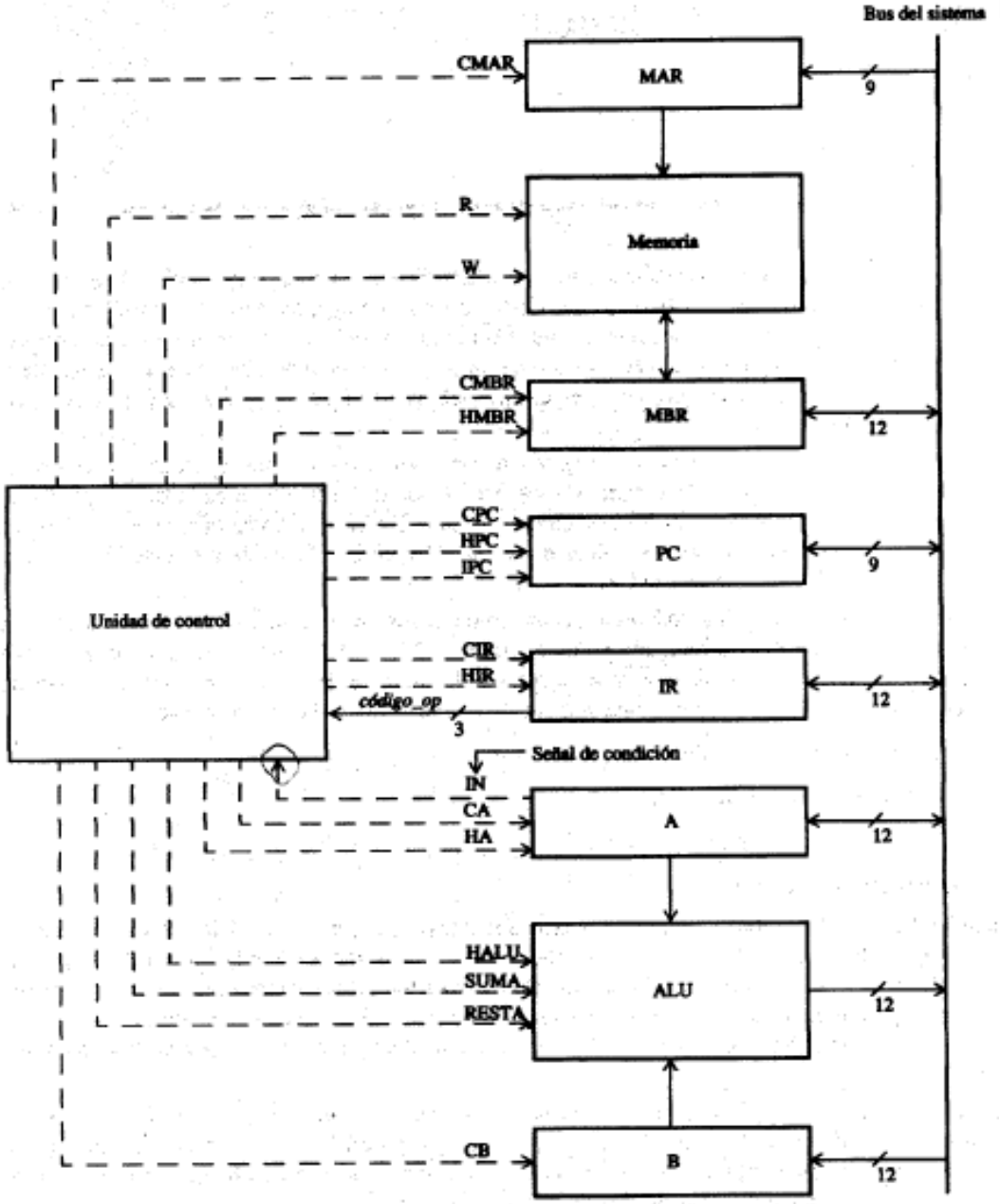


Figura 6.50: Procesador SIMPLE1 y su conexión con la memoria

- 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

- Empleando un procesador de una dirección (procesador con acumulador) con un banco de registro Ri, 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:
- $(((X + Y + Z)X)^{2}) / X = ((X + Y + Z)^{2} X^{2}) / X = (X + Y + Z)^{2} X$
- Respuesta: B= (X= ((X + Y + Z)^{2}) X)

6.6 Introducción a la microprogramación
6.6.1 Modelo original de Wilkes

- Se dice que un procesador está microprogramado cuando las informaciones generadas por la unidad de control se almacenan en una memoria (memoria de control)
  - Cada posición de la memoria de control contiene microinstrucciones
  - Un microprograma es una serie ordenada de microinstrucciones
- Cada instrucción máquina:
  - Un conjunto de microórdenes que especifican las transferencias de información entre los diferentes componentes de la parte operacional
    - ✓ Mediante la activación de los puntos de control, controlamos el flujo de datos
- Microprogramación:
  - Método sistemático para diseñar la unidad de control de cualquier sistema digital

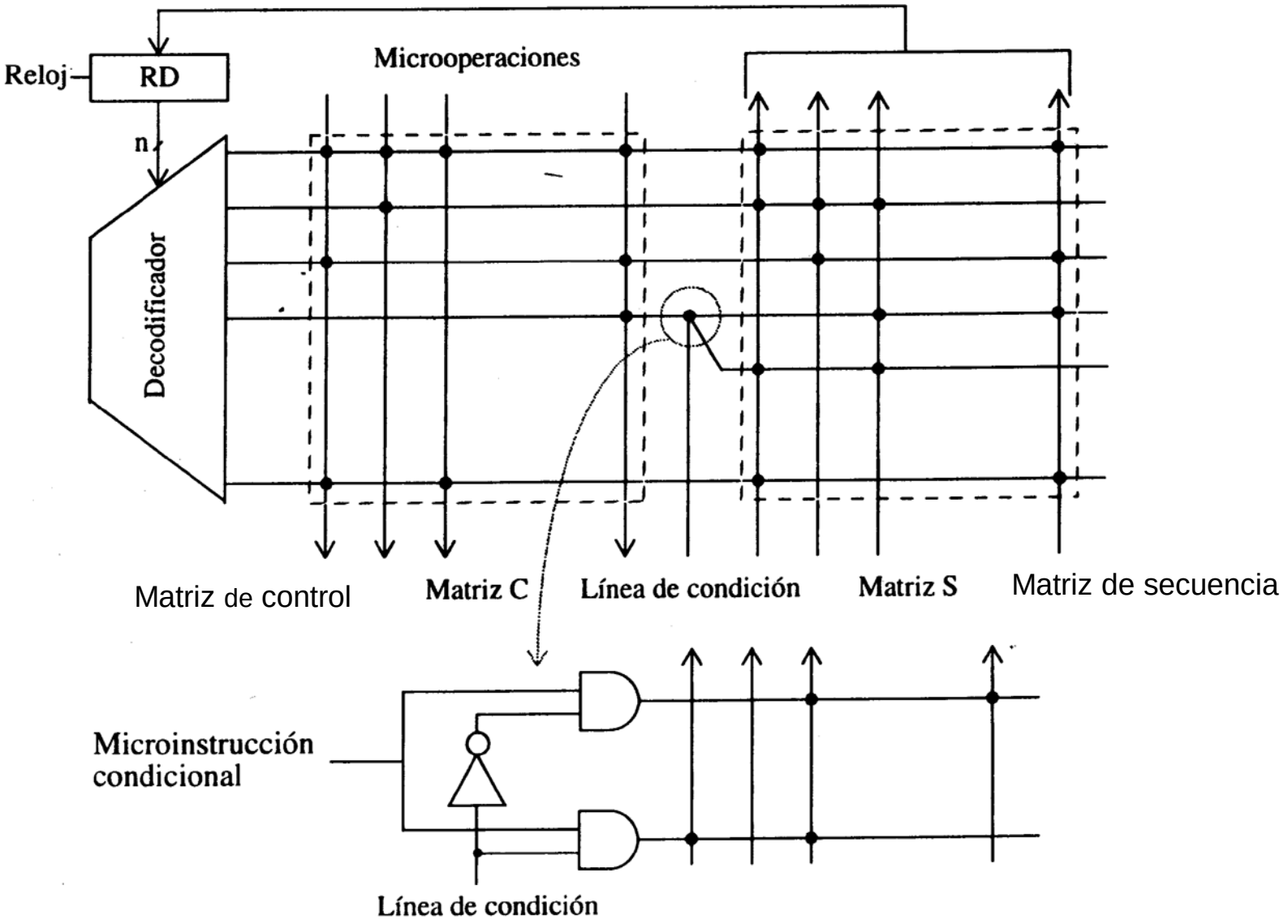


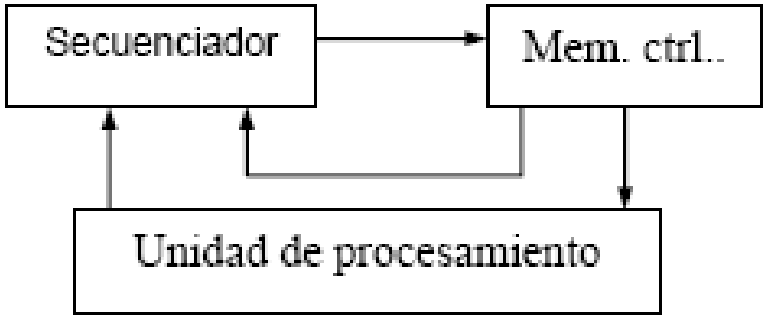
Figura 6-12 Modelo original propuesto por Wilkes

6.6.2 Estructura de una unidad de control microprogramada

- Tareas de la U.C. μ-programada
  - Secuenciamiento de las μ-instrucciones
    - ✓ Obtener la próxima microinstrucción
  - Ejecución de las μ-instrucción
    - ✓ Generar las señales de control
- Una microinstrucción
  - Es un conjunto de micro-órdenes que se pueden ejecutar de forma simultánea y que está contenida en una palabra de la memoria de control

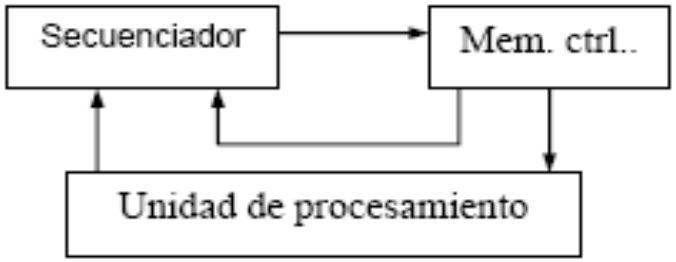
6.6.3 Elementos de una unidad de control microprogramada

- Memoria de control
  - Contiene el micro programa
    - ✓ Micro ordenes
    - ✓ Siguiente micro orden
- Unidad de secuenciamiento
  - Contiene la dirección de la micro orden actual
  - Calcula la siguiente micro orden



Funcionamiento

- 1.- La instrucción entra en IR y tras decodificarse carga en RDC la dirección de la 1ª μ-instrucción.
- 2.- RDC apunta a la memoria de control que saca el dato a RMC
- 3.- RMC contiene 3 campos
  - Señales de control al bus del sistema.
  - Señales de control internas a CPU
  - Próxima dirección de μ-instrucción
- 4.- El secuenciador carga la próxima instrucción en RDC y continúa la secuencia.
  - Opciones:
    - ✓ Bif. a siguiente instrucción:
      - RDC ← RDC+1
    - ✓ Bifurcación: RDC ← RMC [dirección]
      - RDC ← Función (IR[cod. Oper.])



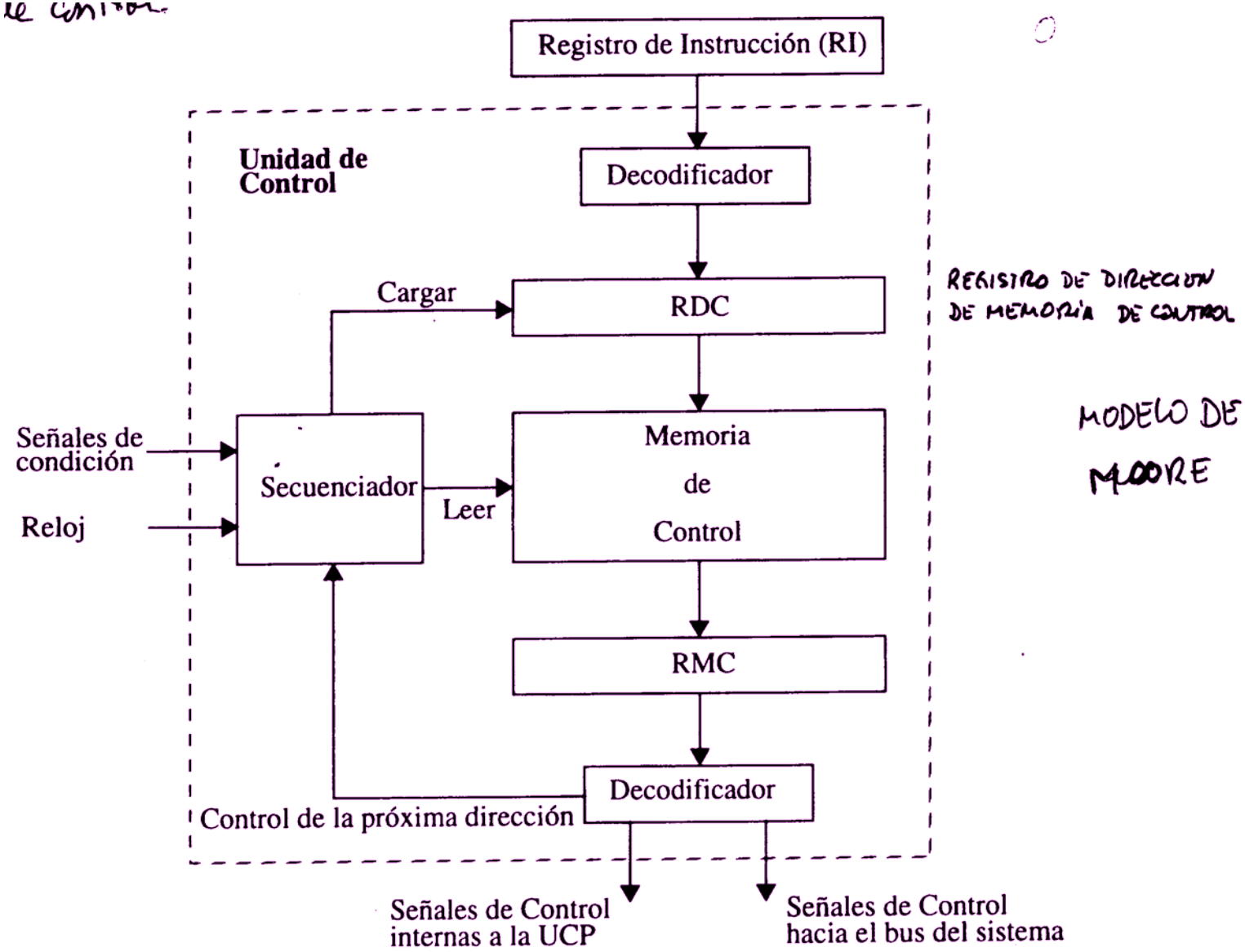
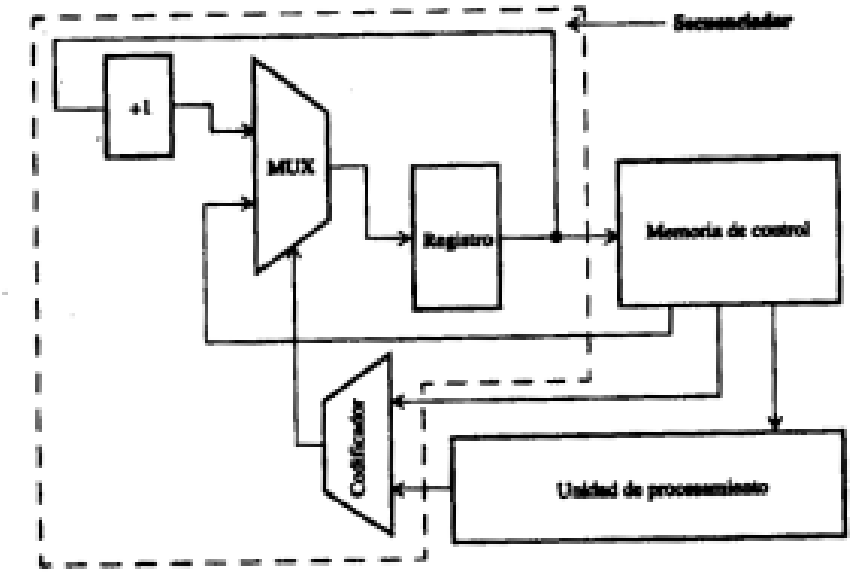


Figura 6-17 Funcionamiento de una Unidad de Control microprogramada

6.6.4 Secuenciamiento de las microinstrucciones

- Se encarga de secuenciar la ejecución de las µ-instrucciones
- Saltos en un secuenciador
  - Salto condicional
  - Incremento de la dirección actual



Secuenciador que permite el salto condicional entre microinstrucciones

Secuenciación

- Para determinar la siguiente microinstrucción:
  - Implícito
    - ✓ Diferencia mediante un campo si se trata de una μ-instrucción de control o de salto.
  - Direccionamiento explícito
    - ✓ Especificado en un campo separado la dirección de salto

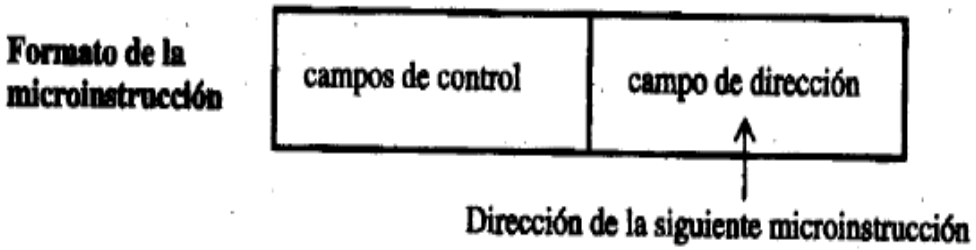
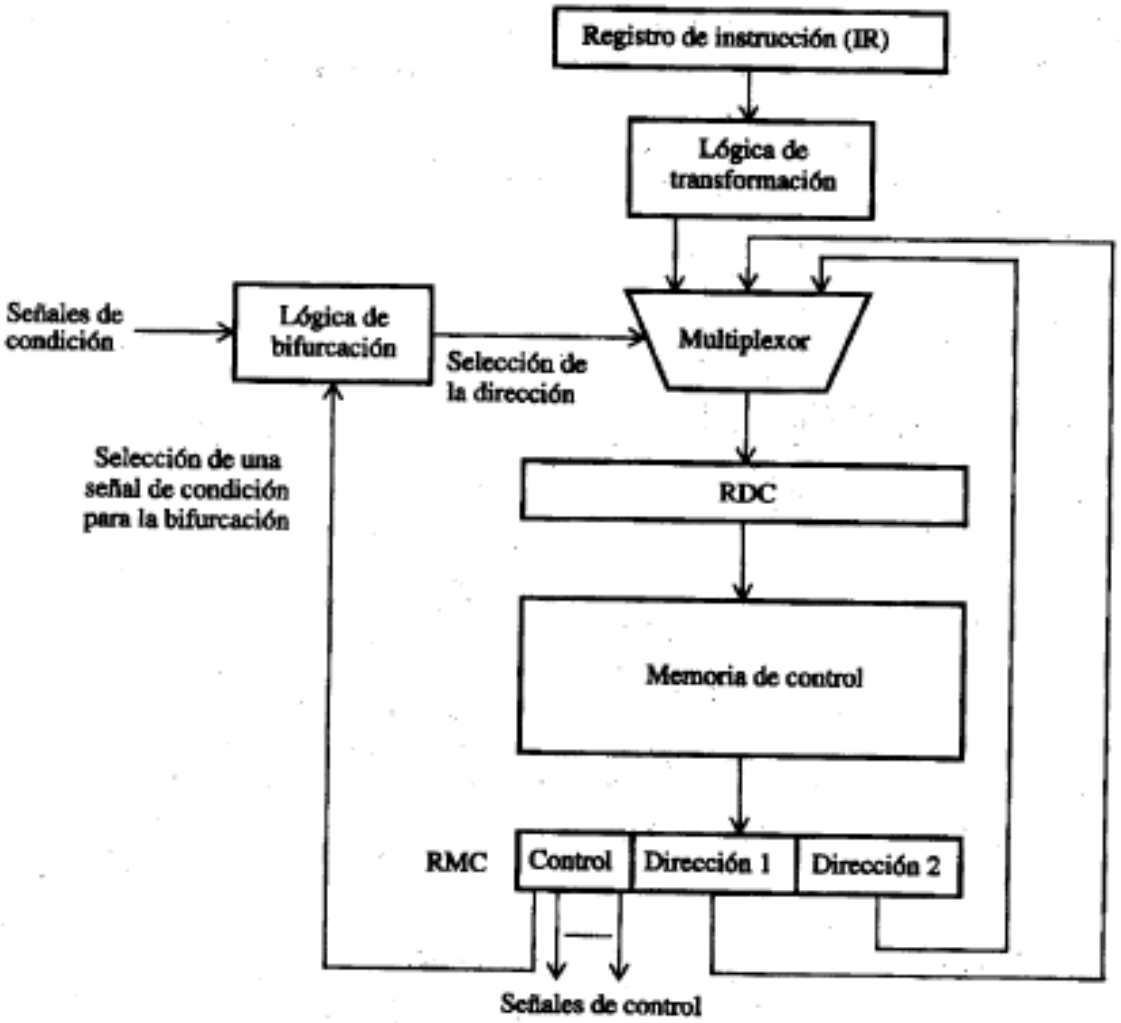


Figura 7.7: Direccionamiento explícito

Direccionamiento explicito con dos direcciones por micro-instrucción



Direccionamiento explicito con una dirección por micro-instrucción

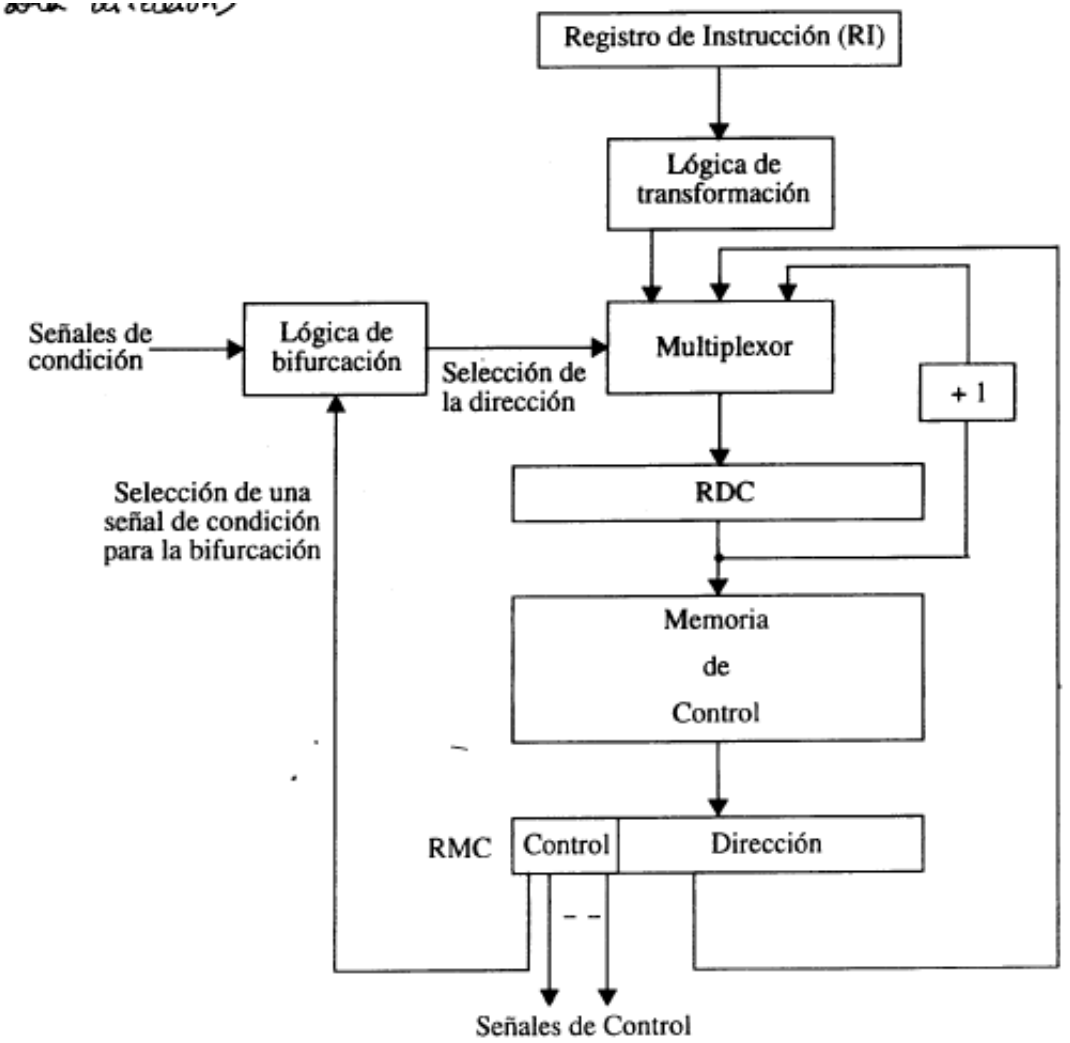
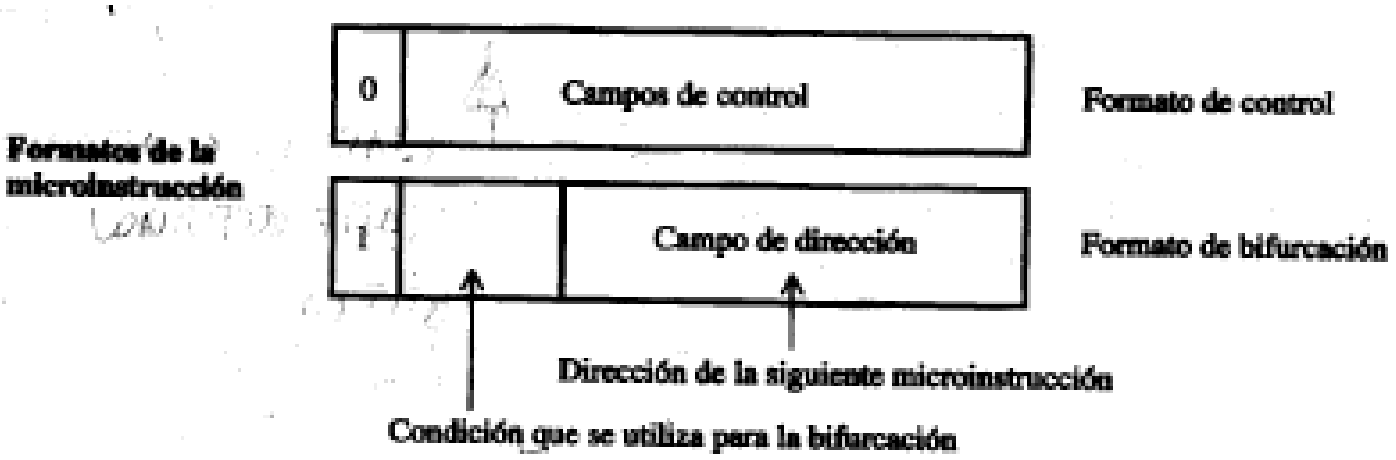
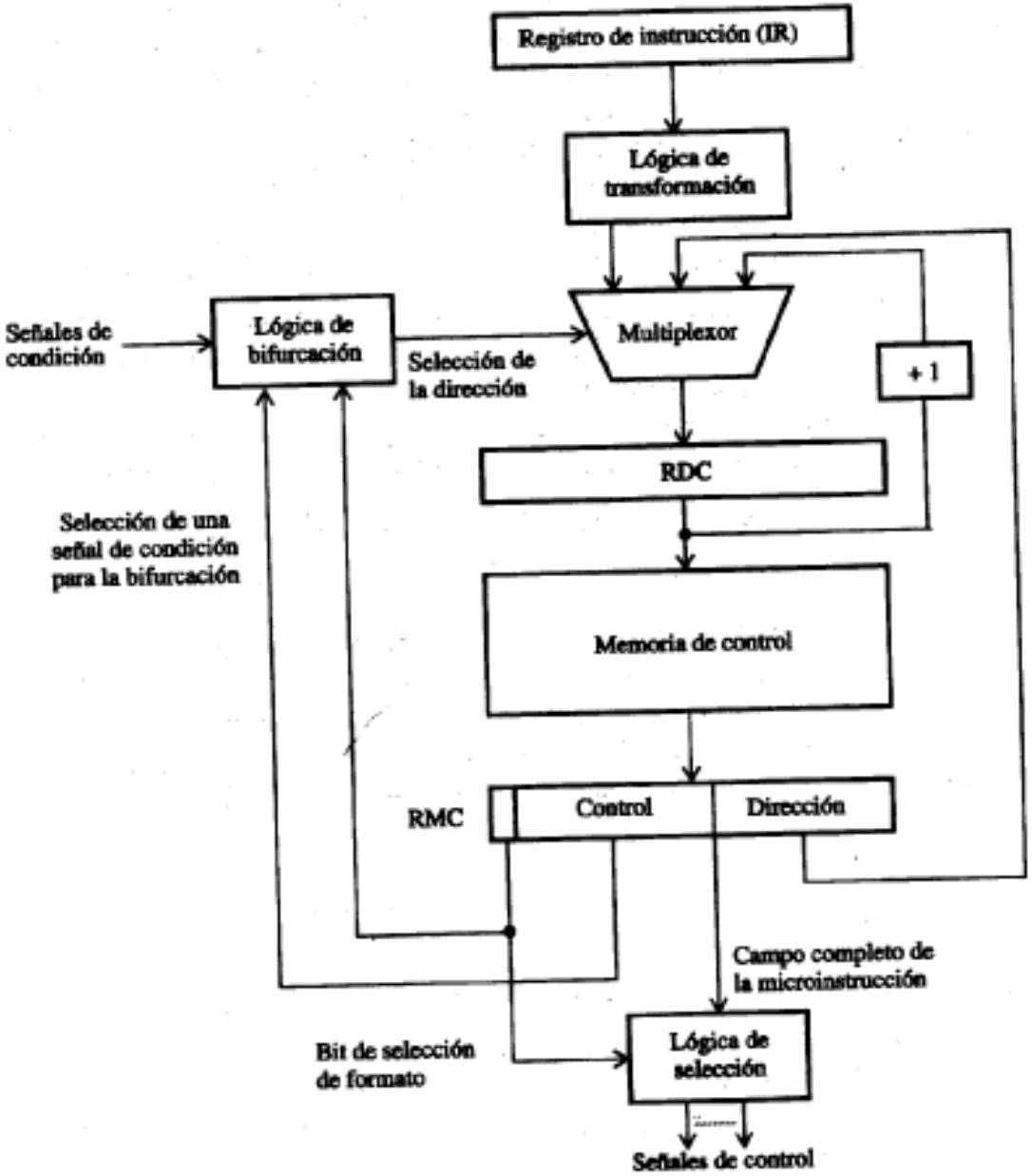


Figura 6-20 Unidad de Control microprogramada con una dirección por microinstrucción

Direccionamiento implícito

- Reduce el tamaño de las microinstrucciones
- Requiere dos tipos (clases) de micro-instrucciones
  - De control
  - De bifurcación
    - ✓ Si cumple la condición bifurca a la dirección especificada
    - ✓ Si no cumple la condición, o es una instrucción de control, sigue en la siguiente instrucción




Registro de instrucción (IR)
Lógica de transformación
Señales de condición
Lógica de bifurcación
Selección de la dirección
Multiplexor
+1
RDC
Selección de una señal de condición para la bifurcación
Memoria de control
RMC
Control
Dirección
Campo completo de la microinstrucción
Bit de selección de formato
Lógica de selección
Señales de control

6.6.5 Organización de la memoria de control

- Una micro-instrucción tiene dos partes:
  - Bits de la dirección
  - Bits correspondientes a las señales de control
    - ✓ Vamos a ver como podemos representar los bits de las señales de control
    - ✓ Dos opciones
      - ➢ Microprogramación horizontal
      - ➢ Microprogramación vertical
- Microprogramación horizontal
  - Las señales de control se representan Sin codificar
  - Un bit por cada señal de control que exista en la unidad de control
    - ✓ La mayoría estarán a cero (solo activa las que afecten a esa micro-instrucción)
    - ✓ Tamaño muy grande

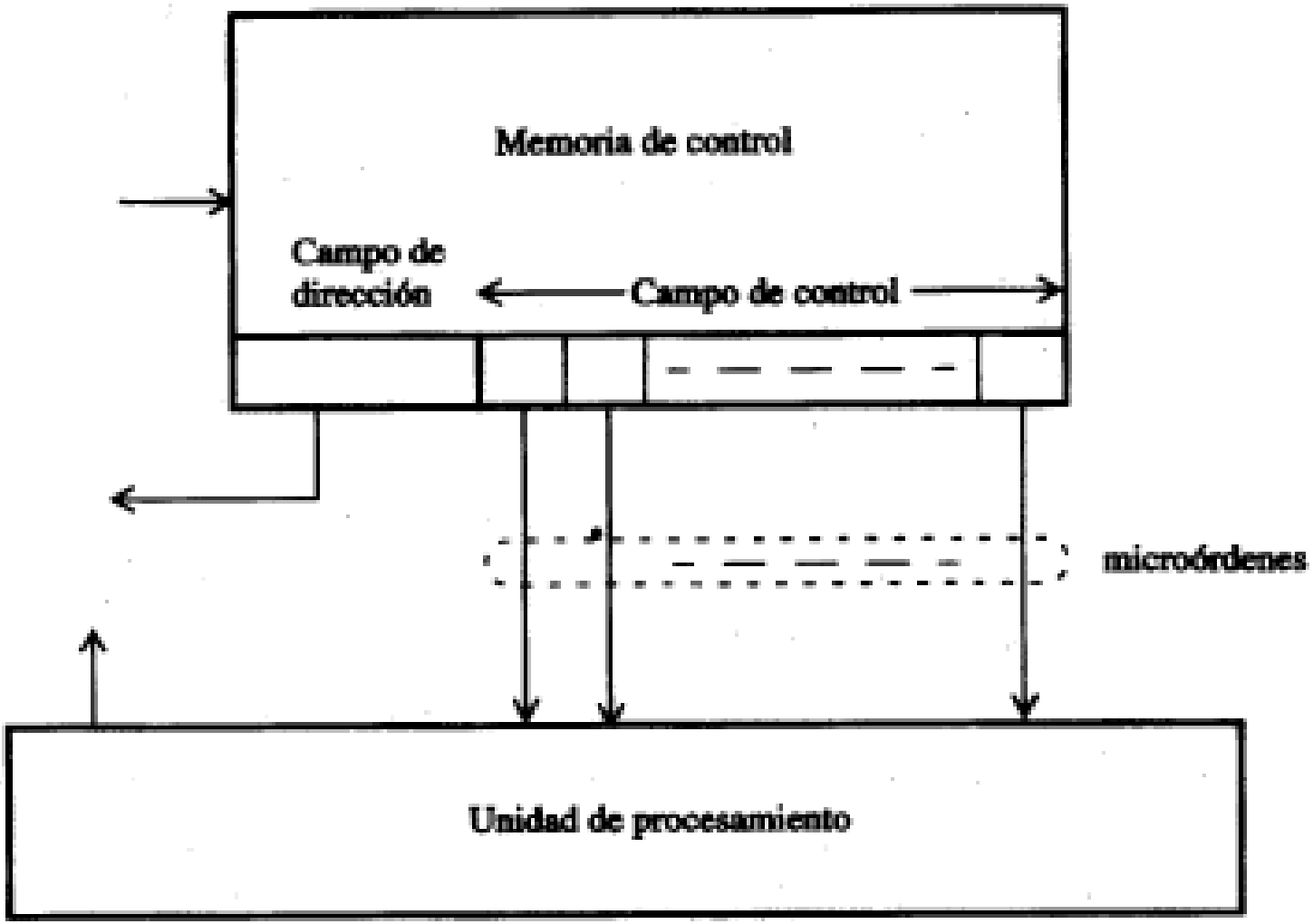


Figura 7.12: Formato horizontal de una microinstrucción

Microprogramación vertical

- Se codifican las señales (se numeran y se identifican por su número)
- El campo de control está dividido en subcampos
- Cada uno de los subcampos controla operadores excluyentes entre si (no se pueden producir en el mismo instante)
  - Los subcampos de control contiene la identificación de la señal de control
  - Con subcampos de j bits se pueden especificar 2^j-1 señales. Una codificación está reservada para cuando ninguna está activa
- El formato vertical es más lento porque precisa la decodificación de las señales

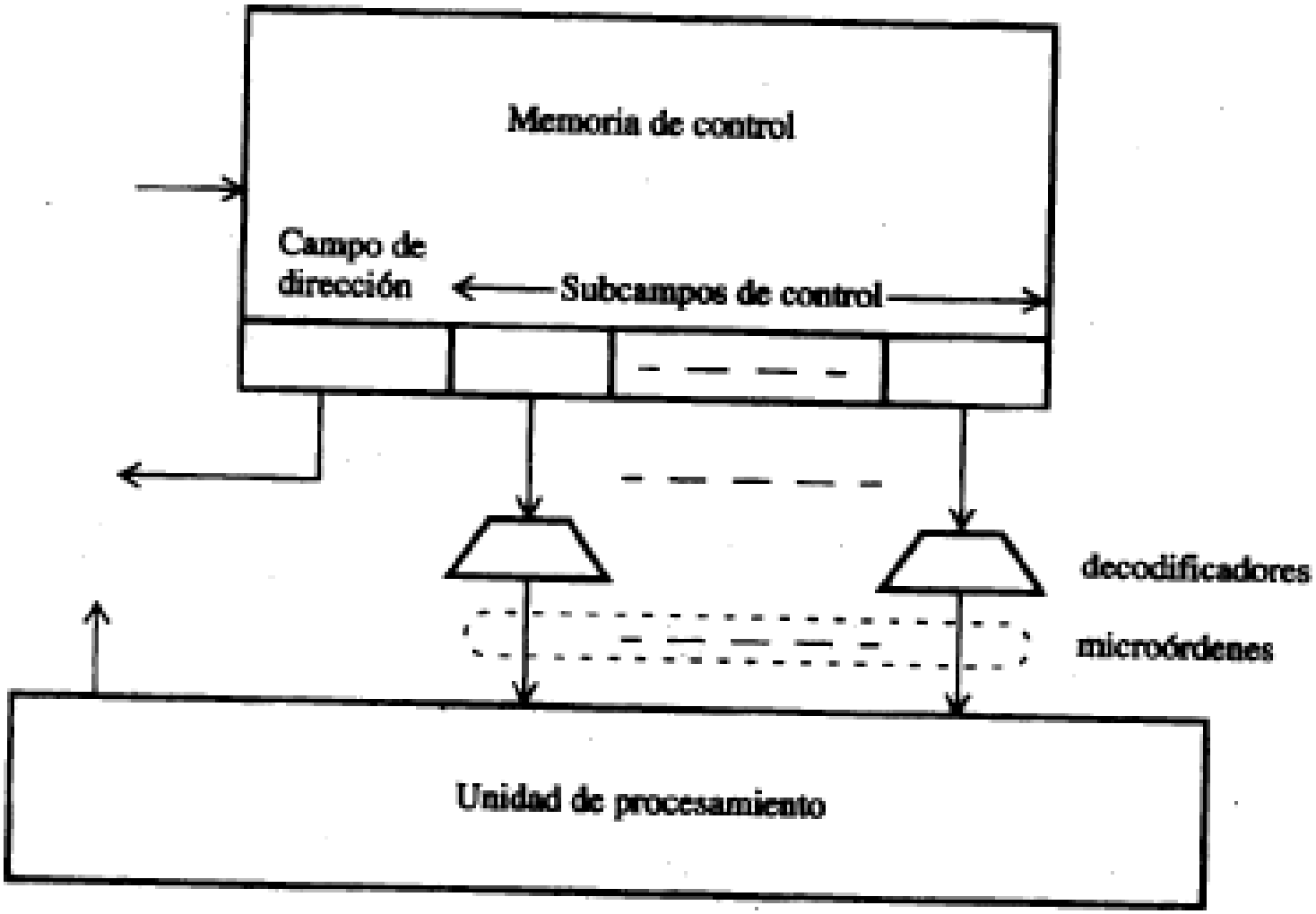


Figura 7.13: Formato vertical de una microinstrucción codificada por subcampos

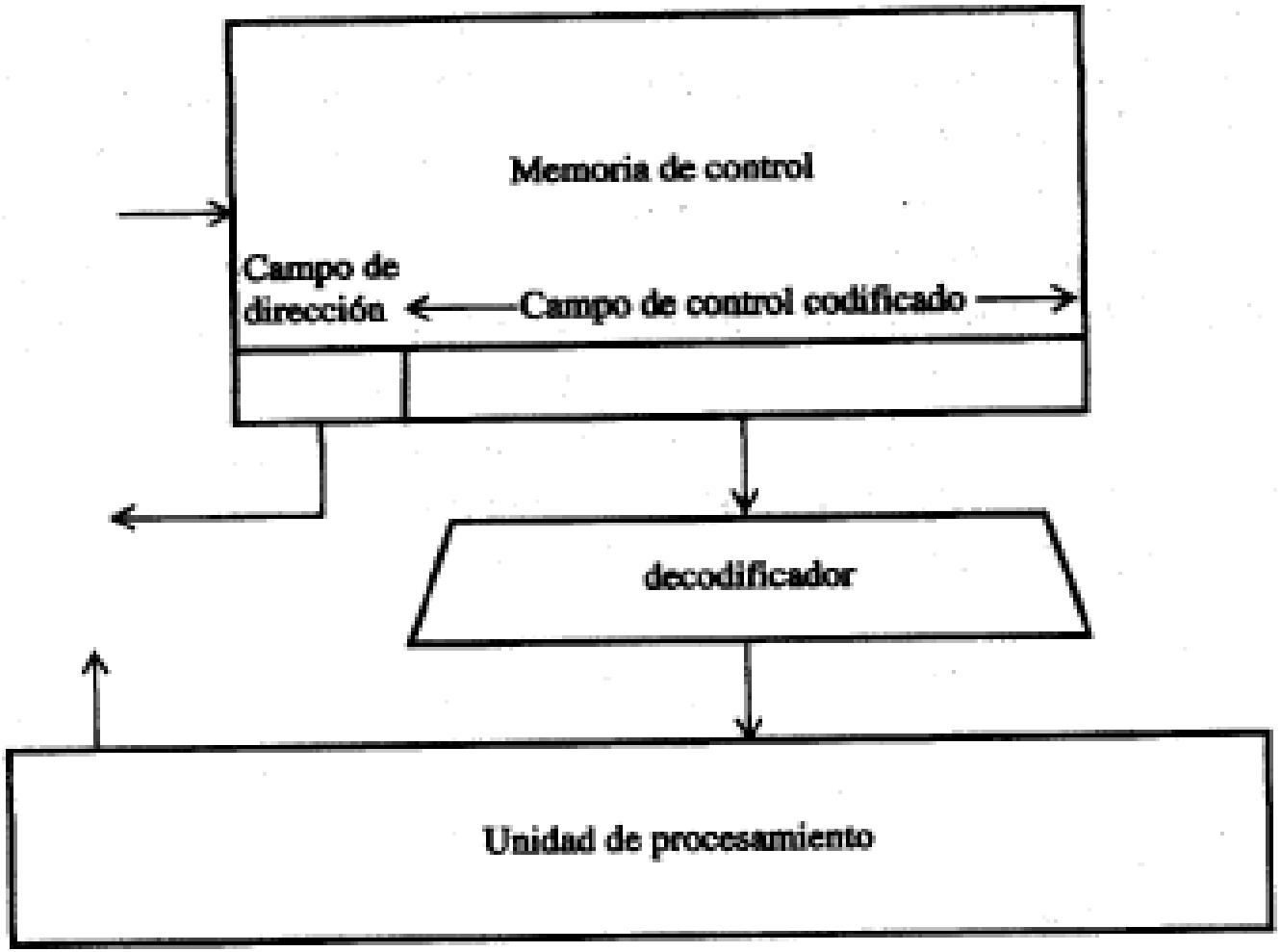


Figura 7.14: Formato vertical de una microinstrucción totalmente codificada

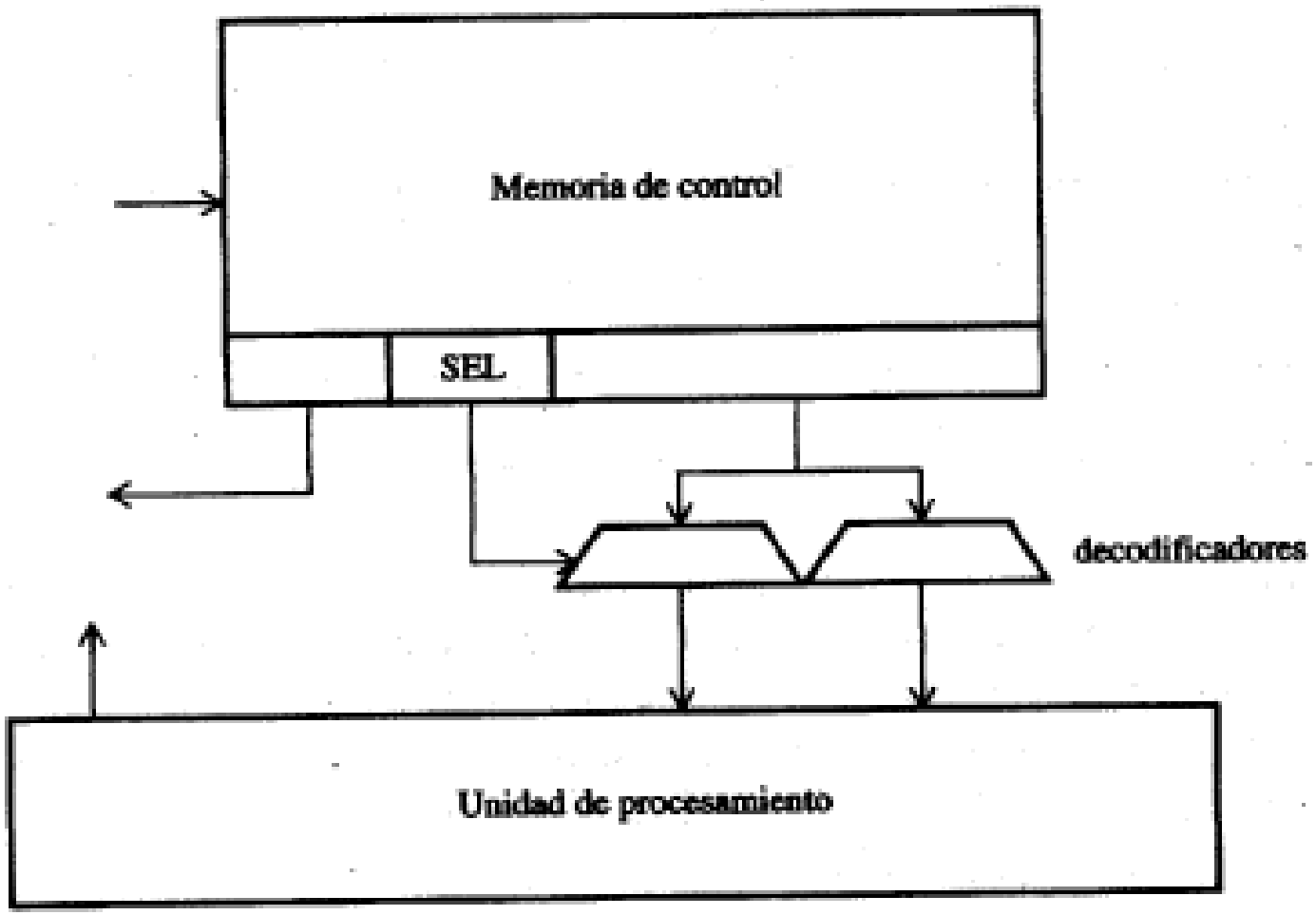


Figura 7.15: Formato vertical con campo de selección del tipo de microinstrucción

AVISO

- ***Mucho cuidado porque en el caso del formato vertical, al codificar las señales siempre hay que tener en cuenta una codificación de más, la correspondiente a que todas las señales de control estén desactivadas.***
- ***El caso del formato horizontal, esto no se considera porque cuando se precisa que no esté activada ninguna de las señales, basta con ponerlas todas a 0.***

6.6.6 Ejecución de las microinstrucciones

- Fases
  1. Búsqueda de la μ-instrucción
  2. Decodificación de los campos de la μ-instrucción
  3. Ejecución de las μ-operaciones
  4. Cálculo de la dirección de la próxima μ-instrucción
- Tipos de ejecuciones de las microinstrucciones
  - Monofásicas
    - ✓ Las μ-operaciones se pueden ejecutar al mismo tiempo
  - Polifásicas
    - ✓ Los campos se utilizan de forma escalonada en el tiempo de ejecución

- Una unidad de control micro-programada con direccionamiento implícito tiene una memoria de control con 24 bits de longitud de palabra.
- Si las microinstrucciones emplean 15 bits para los campos de control y el tamaño máximo de la memoria de control de esta Unidad de Control microprogramada es de 2^{20} palabras
- ¿Cuántas condiciones como máximo se pueden evaluar en el caso de microinstrucciones de bifurcación?
  - A) 3
  - B) 16
  - C) 4
  - D) Ninguna de las anteriores.

SOLUCIÓN

- [Ver sección 6-2 del libro de teoría.]
- En el caso de direccionamiento implícito, se definen dos formatos de instrucciones diferenciadas por el primer bit.
- En caso de ser 0 tienen formato de control y se usan 15 bits adicionales para el control, es decir, se usan 16 bits y quedan 8 libres sin usar.
- En el caso de formato de bifurcación (primer bit a 1) se debe evaluar la condición que sea necesaria. Como el número de palabras de 2^{20}, 20 es número de bits necesarios para el campo dirección.
- Por lo tanto, el número de bits utilizados para codificar el número máximo de condiciones de bifurcación será: 24-20-1= 3 bits. Siendo por lo tanto 8 el número máximo de condiciones.
- Respuesta: D

- El tamaño máximo de la memoria de control de un computador es de 2^{20} palabras.
- La parte de control del formato de una microinstrucción (de la unidad de control microprogramada) de este computador emplea 8 bits para seleccionar independientemente las microoperaciones que se llevan a cabo.
- Esta unidad de control tiene un campo para indicar el tipo de bifurcación a realizar (condicional, incondicional o no bifurcar) y un campo para seleccionar una única señal de condición de las 32 señales disponibles.
- ¿Cuál es el tamaño de las microinstrucciones en bits?
  - A) 20
  - B) 35
  - C) 52
  - D) Ninguna de las anteriores

SOLUCIÓN

- [Ver problema 7.2 del libro de problemas].
- Como el tamaño máximo de la memoria de control es de 20^{20} palabras, se necesitan reservar 20 bits para la dirección.
- Si la microinstrucción tiene X bits de longitud, de los cuales 8 bits se emplean para seleccionar las microoperaciones,
- 2 bits para codificar el tipo de bifurcación y
- 5 (2^5=32) para seleccionar una única señal de condición,
- X=20+8+2+5=35 De esta forma, el tamaño de las microinstrucciones es de 35 bits.
- Respuesta: B (35)

- Un computador usa un formato de microinstrucción mixto, parte horizontal y parte vertical.
- La parte con formato horizontal de codificación tiene una longitud de *k* bits y
- La parte con formato vertical de codificación posee *m* campos codificados de *n* bits cada uno.
- ¿Cuál es el máximo número de señales de control que pueden usarse en este computador?
  - A) $k + n \times 2^m$
  - B) $k + n^m$
  - C) $k + n\ x\ (2^m\text{-}1)$
  - D) Ninguna de las anteriores.

Solución

- La organización de la memoria de control se explica en el Apartado 7.2.4 del texto de teoría.
- La parte con *formato horizontal* de codificación tiene una longitud de *k* bits, por tanto pueden usarse *k* señales de control.
- La parte con *formato vertical* de codificación posee *m* campos codificados de *n* bits cada uno.
- Cada campo de *n* bits codifica 2^n-1 señales.
- Así pues, el número máximo de señales de control que puede usarse en este computador es $k+m\times(2^n-1)$.
- Respuesta:
  - D (Ninguna de las anteriores)

- En una Unidad de Control microprogramada con formato de microinstrucciones *vertical*,
- Un subcampo que deba especificar 16 señales de control
- Habrá de tener una anchura mínima de:
  - A) 4 bits.
  - B) 5 bits.
  - C) 16 bits.
  - D) Ninguna de las anteriores
- SOLUCIÓN
  - Un subcampo codificado de j bits puede especificar a lo sumo 2j-1 señales de control.
  - Para gobernar 16 señales de control no es suficiente con 4 bits ya que 24-1 = 15 < 16, pero si son suficientes 5 bits: 25-1 = 31
  - Respuesta correcta:
    - ✓ B(5 bits)

- Un computador usa el formato vertical de codificación de instrucciones para parte de las señales de control y el formato horizontal para *k* señales de control.
- El formato vertical posee *n* campos codificados de *m* bits cada uno.
- ¿Cuál es el máximo número de señales de control que pueden usarse en este computador?
  - A) $k + nx2^m$.
  - B) $k + n^m$.
  - C) $k + nx(2^m-1)$.
  - D) Ninguna de las anteriores.
- SOLUCION
  - Respuesta correcta: C ($k + nx(2^m-1)$).

- Un computador microprogramado tiene un total de 132 señales de control.
- De ellas, un grupo de 16 son mutuamente excluyentes entre sí y otro grupo de 30 son mutuamente excluyentes entre sí.
- Indique si las siguientes afirmaciones son verdaderas:
- I. Utilizando formato vertical, el tamaño de los subcampos codificados sería de 4 y 5 bits, respectivamente.
- II. Al existir señales mutuamente excluyentes no puede utilizarse el formato vertical de microinstrucciones.
  - A) I: sí, II: sí.
  - B) I: sí, II: no.
  - C) I: no, II: sí.
  - D) I: no, II: no.
- SOLUCIÓN
  - La afirmación I es falsa ya que un subcampo codificado de i bits puede codificar a lo sumo 2^i-1 señales de control diferentes. Por este motivo harían falta dos subcampos de 5 bits cada uno.
  - La afirmación II es falsa ya que el formato vertical de microinstrucciones saca partido precisamente del hecho de que existan señales de control mútuamente excluyentes.
  - Respuesta correcta:
    - ✓ D (I: no; II: no).

- Una Unidad de Control microprogramada con direccionamiento explícito con dos direcciones por microinstrucción, tiene una memoria de control con 35 bits de longitud de palabra.
- Si las microinstrucciones emplean 15 bits para los campos de control, el número máximo de palabras de la memoria de control de esta Unidad de Control microprogramada es de:
  - A) 2^{10} palabras
  - B) 2^{20} palabras
  - C) 2^{17} palabras
  - D) Ninguna de las anteriores

- SOLUCIÓN
  - [Ver apartado 7.2.3 del texto base de teoría y los problemas 7-1 y 7-2 del libro de problemas].
  - De los 35 bits, como 15 son para los campos de control, quedan 20 bits para las dos direcciones, 10 para cada una de ellas.
  - Por tanto, el tamaño máximo de la memoria de control es de 2^{10} palabras.
  - Respuesta:
    - ✓ A (2^{10} palabras)

Ejemplos de la edición anterior

FIN DEL CAPÍTULO

Ejemplos:
Desarrollo de un sencillo microprograma

- Diseño de la Unidad de Control de un procesador sencillo con las siguientes especificaciones:
- Elementos:
  - M[MAR] → memoria RAM de 256*8 bits
  - T → Registro de 8 bits
  - R1 → Registro de 8 bits
  - MAR → Registro de dirección de Memoria de 8 bits
  - DECR (-1) → operador de Decremento en 1
  - INCR (+1) → operador de Incremento en 1
  - SUM → Sumador

Microórdenes de la ruta de datos		
Microorden	Significado	Acción
µord0	Decrementa T	T ← T-1
µord1	Transfiere T+1 a MAR	MAR ← T+1
µord2	Transfiere T a MAR	MAR ← T
µord3	Suma R1 a M[MAR]	M[MAR] ← M[MAR]+R1
µord4	Transfiere M[MAR] a R1	R1 ← M[MAR]

Microórdenes para el secuenciamiento		
Microorden	Significado	Acción
µord5	Secuencia normal	RDC ← RDC+1
µord6	Salto condicional a µI(x)	RDC ← x

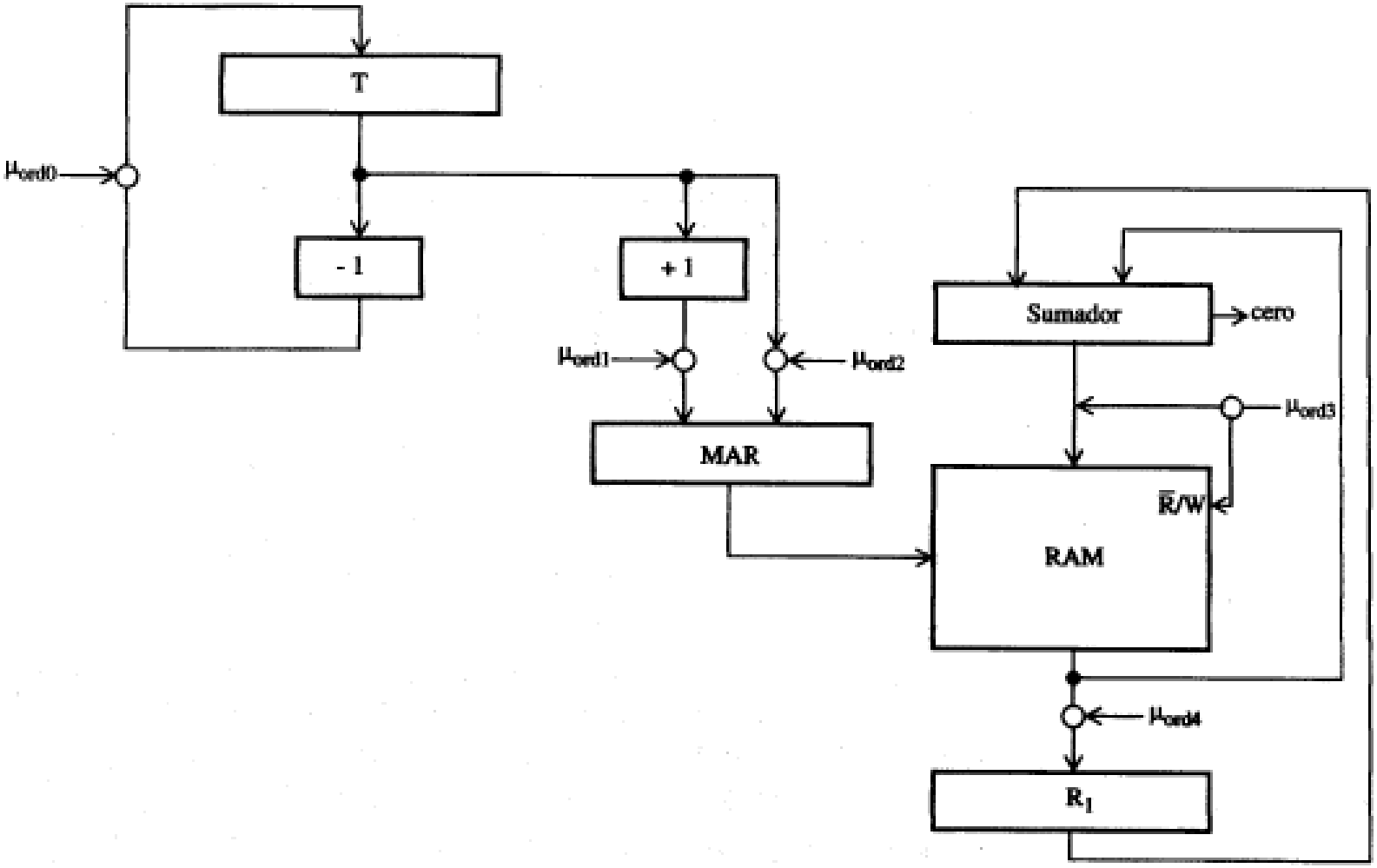


Figura 7.21: Ruta de datos del procesador del ejemplo 7.3.1

Objetivo del microprograma

- T ← T-1
- M[T] ← M[T] + M[T+1]
- **If** M(T) = 0 **then** RDC ← x **else** RDC ← RDC+1

- Unidad de control

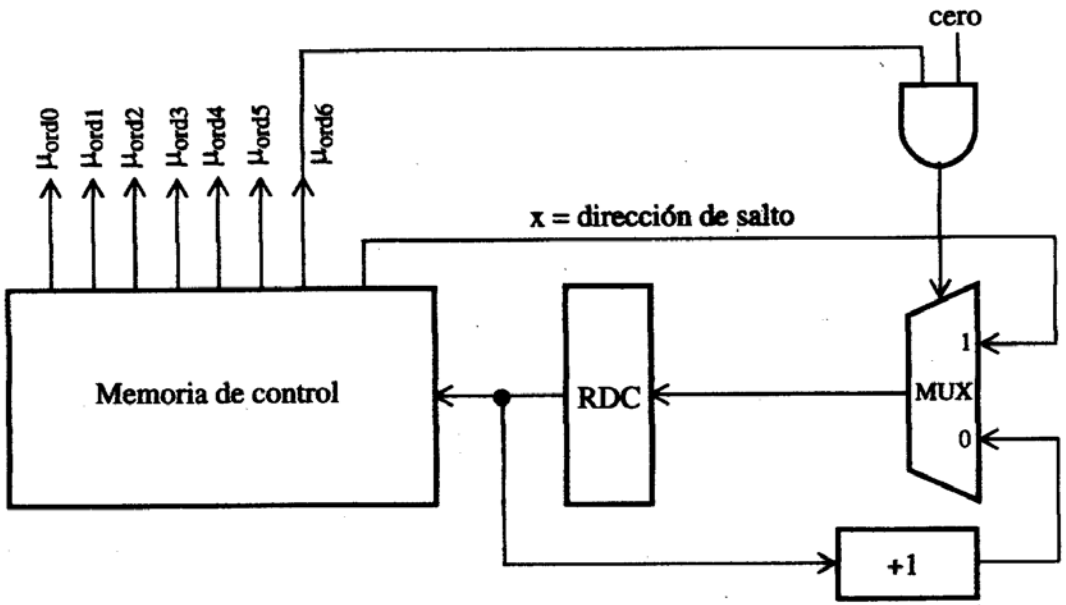


Figura 7.22: Unidad de control del procesador del ejemplo 7.3.1

Primera solución: Formato horizontal

- Cada señal de control, tiene su propio bit en la memoria :
  - Mayor anchura de la memoria **15 bits**

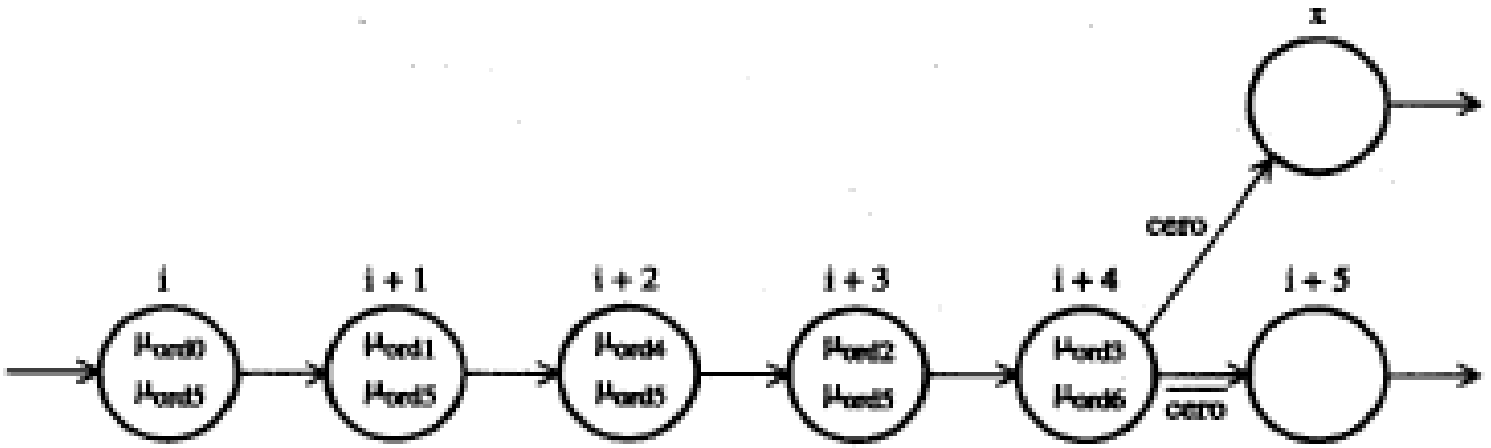


Figura 7.23: Grafo orientado del microprograma para la primera solución del ejemplo 7.3.1

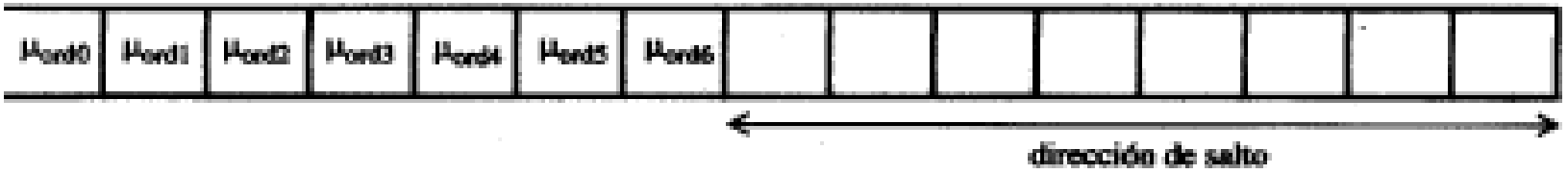


Figura 7.24: Formato de la microinstrucción para la primera solución del ejemplo 7.3.1

En la Tabla 7.3 se muestra el código binario del microprograma correspondiente a esta solución que está upuesto de 5 palabras de 15 bits.

Microprograma	Codificación binaria							
	Pord0	Pord1	Pord2	Pord3	Pord4	Pord5	Pord6	dirección de salto
μI (i)	1	0	0	0	0	1	0	- - - - - - - -
μI (i + 1)	0	1	0	0	0	1	0	- - - - - - - -
μI (i + 2)	0	0	0	0	1	1	0	- - - - - - - -
μI (i + 3)	0	0	1	0	0	1	0	- - - - - - - -
μI (i + 4)	0	0	0	1	0	0	1	x

Tabla 7.3: Código binario del microprograma para la primera solución del ejemplo 7.3.1

Segunda solución: Formato vertical

- El campo de control está dividido en subcampos
- Cada subcampo controla un conjunto de operadores
  - Estos operadores son excluyentes entre sí (no se pueden producir en el mismo instante)
  - Está codificado indicando la señal a controlar
  - Se precisa un decodificador por subcampo

μ_1	μ_2	μ_3	microorden
0	0	0	μ_{ord0}
0	0	1	μ_{ord1}
0	1	0	μ_{ord2}
0	1	1	μ_{ord3}
1	0	0	μ_{ord4}
1	0	1	-
1	1	0	-
1	1	1	-

Codificación de las microórdenes μ_{ord0}, μ_{ord1}, μ_{ord2}, μ_{ord3} y μ_{ord4}

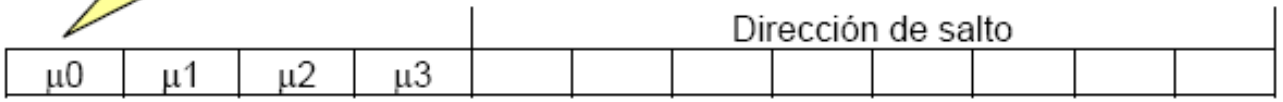
Según secuencia del grafo

Microprograma	Codificación binaria											
	μ_0	μ_1	μ_2	μ_3	dirección de salto							
µI (i)	0	0	0	0	-	-	-	-	-	-	-	-
µI (i + 1)	0	0	0	1	-	-	-	-	-	-	-	-
µI (i + 2)	0	1	0	0	-	-	-	-	-	-	-	-
µI (i + 3)	0	0	1	0	-	-	-	-	-	-	-	-
µI (i + 4)	1	0	1	1	x							

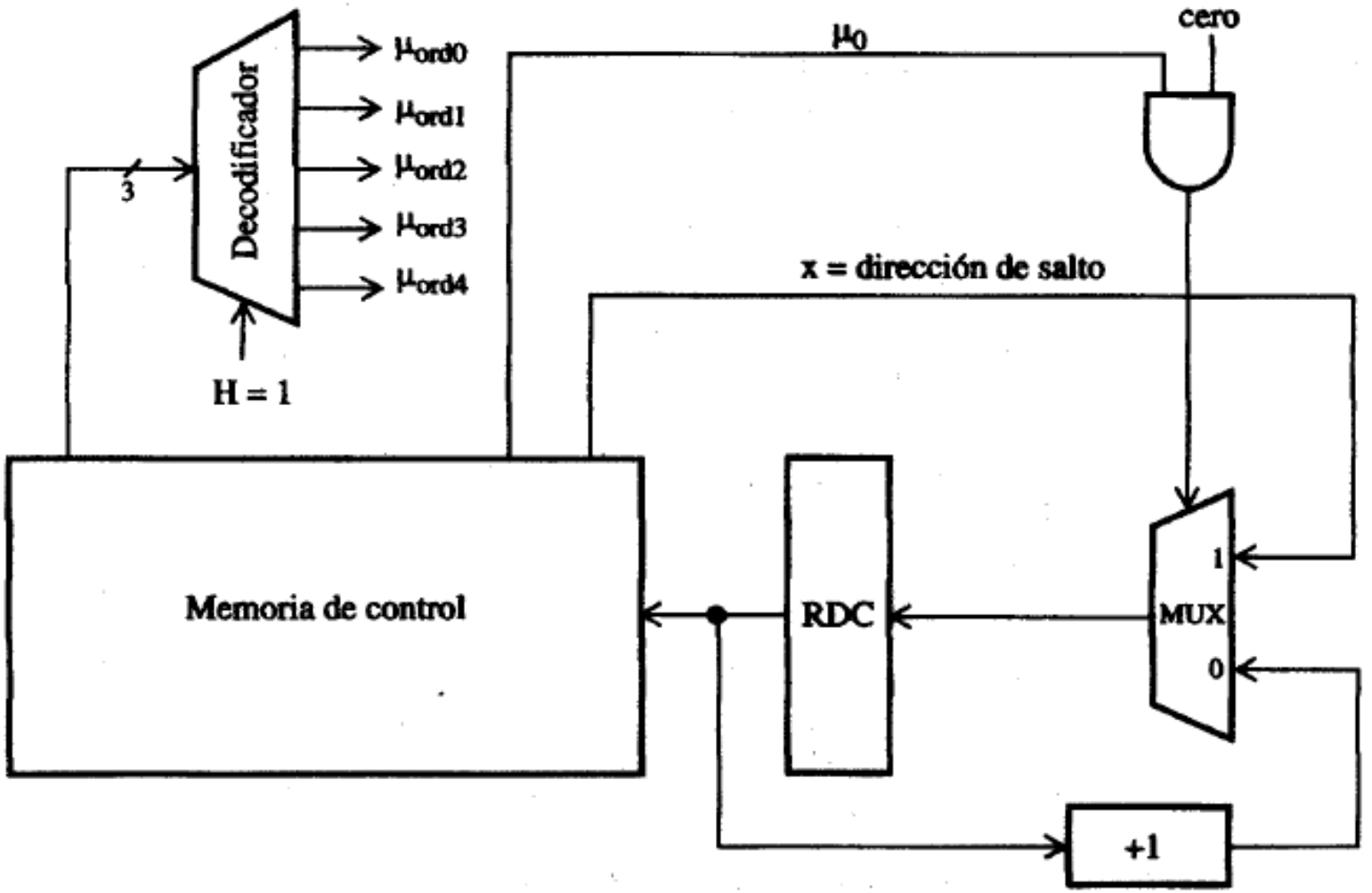
Código binario del microprograma

µ0rd5 ⇒ µ0=0
µ0rd6 ⇒ µ0=1

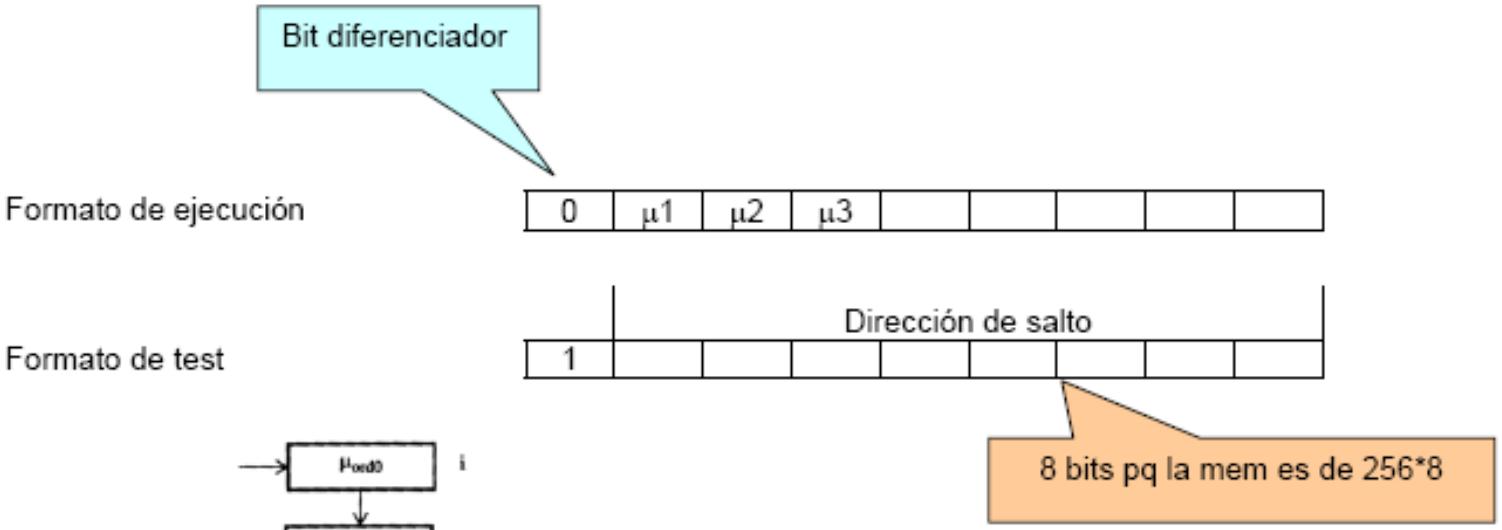
Formato de la µI:

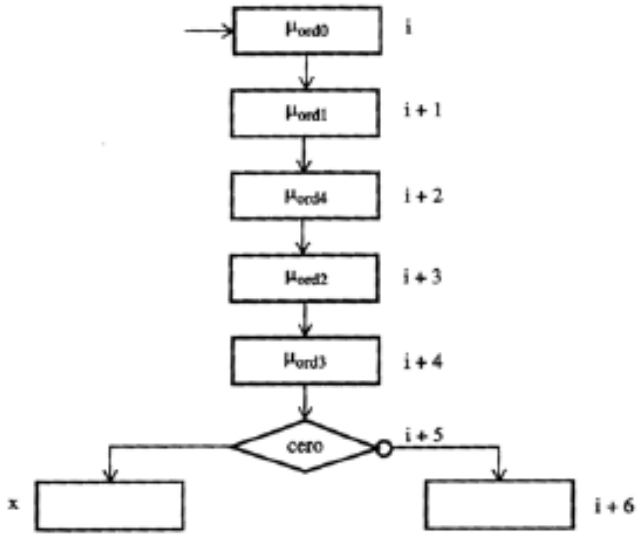


Hay un decodificador adicional



Tercera solución: Direccionamiento implícito microprogramación vertical





Organigrama equivalente al grafo orientado

Microprograma	Codificación binaria								
	OP	μ1	μ2	μ3					
μi (i)	0	0	0	0	-	-	-	-	-
μi (i + 1)	0	0	0	1	-	-	-	-	-
μi (i + 2)	0	0	1	0	-	-	-	-	-
μi (i + 3)	0	0	1	0					
μi (i + 4)	0	0	1	1	-	-	-	-	-
μi (i + 4)	1	x							

Código binario del microprograma

Microprograma	Codificación binaria								
	OP	μ_1	μ_2	μ_3					
μI (i)	0	0	0	0	-	-	-	-	-
μI (i + 1)	0	0	0	1	-	-	-	-	-
μI (i + 2)	0	1	0	0	-	-	-	-	-
μI (i + 3)	0	0	1	0					
μI (i + 4)	0	0	1	1	-	-	-	-	-
μI (i + 4)	1					x			

Tabla 7.6: Código binario del microprograma para la segunda solución del ejemplo 7.3.1

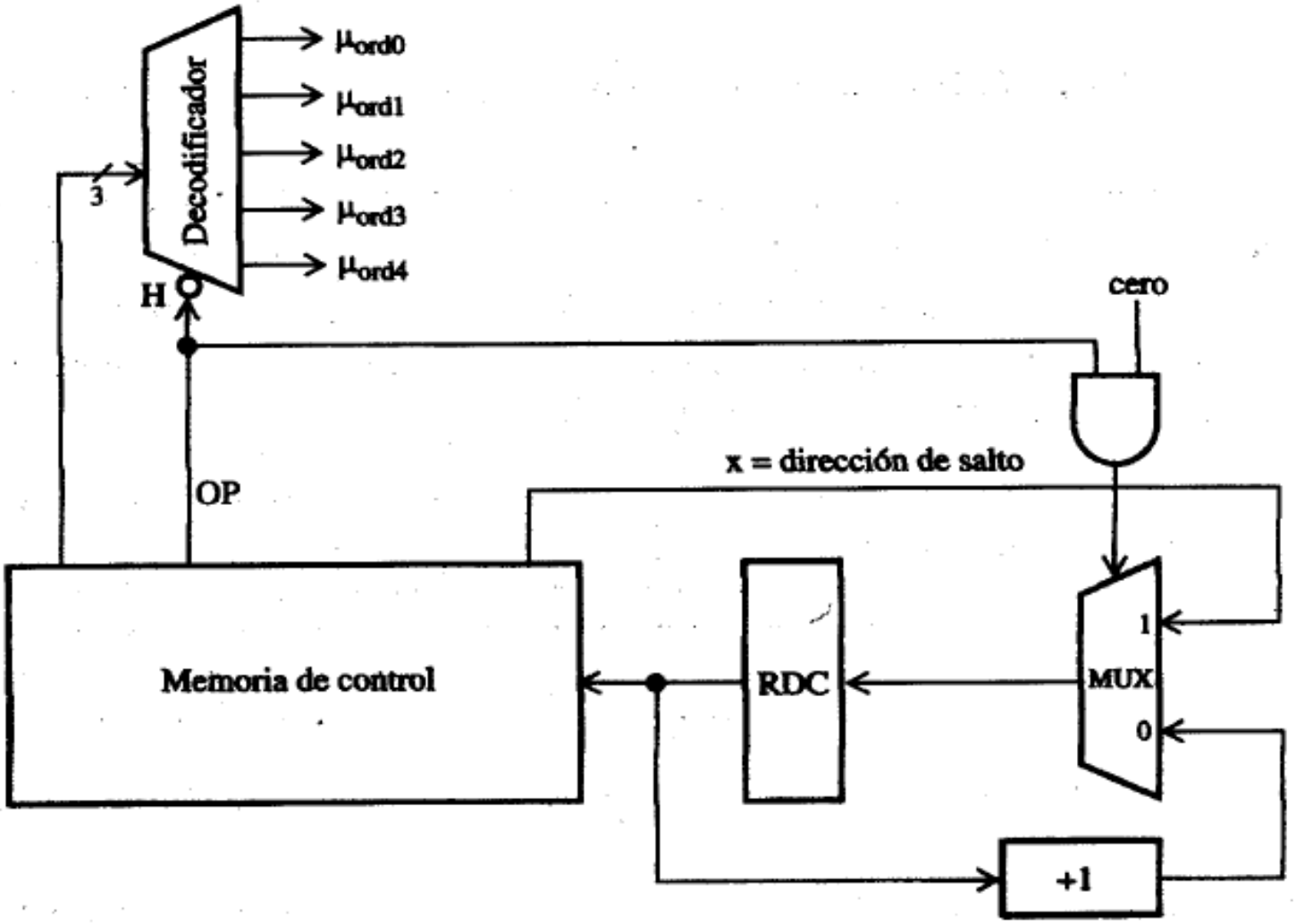


Figura 7.29: Esquema de la unidad de control para la tercera solución del ejemplo 7.3.1

Cuarta solución

- Se basa en reordenar la secuencia de μ-ordenes para realizar el mismo tiempo las que no tengan influencia unas sobre otras.

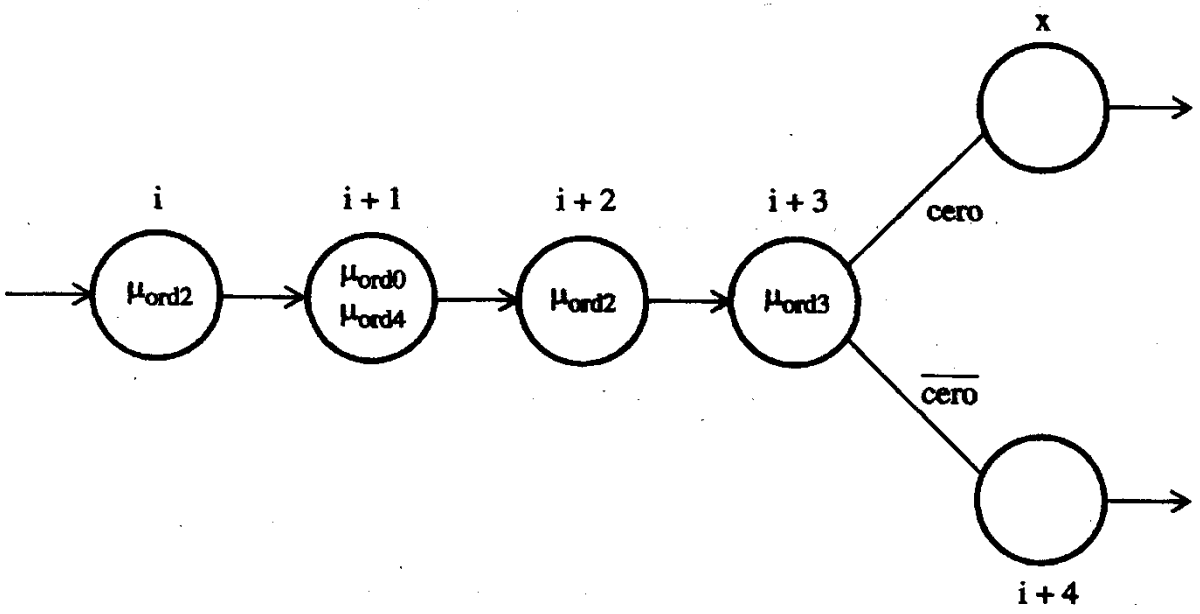


Figura 7.30: Grafo orientado del microprograma para la cuarta solución del ejemplo 7.3.1

7.4 Ejemplo de diseño micro-programado: multiplicador binario

1) *Inicializar* y *Borrar_C* en el estado S_{00}
2) *Sumar_Cargar* en el estado S_{10}
3) *Borrar_C* y *Desplazar_Dec* en el estado S_2
4) Ninguna señal de control activa en los estados S_0 y S_1

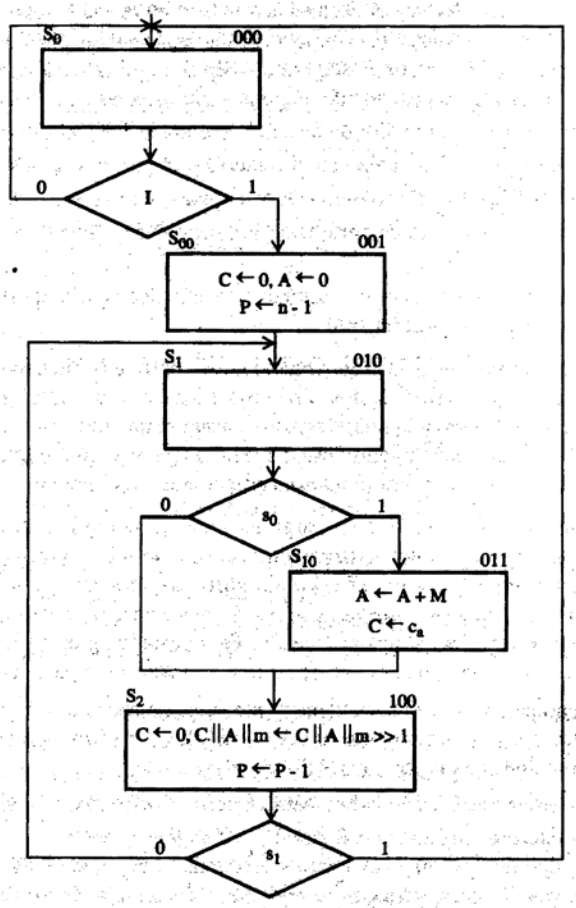


Figura 7.31: Diagrama ASM para el diseño de la unidad de control microprogramada del multiplicador

Borrar_C. Estas señales de control y las transferencias de registros asociadas se muestran en la Tabla 7.7 .

Señal de control	Transferencia de registro	Estados en los que la señal está activa	Bit de posición en la µI	Nombre simbólico
Inicializar	$A \leftarrow 0, P \leftarrow n-1$	S_{00}	0	IN
Sumar_Cargar	$A \leftarrow A + M, C \leftarrow c_a$	S_{10}	1	SC
Borrar_C	$C \leftarrow 0$	S_{00}, S_2	2	BC
Desplazar_Dec	$C \Vert A \Vert m \leftarrow C \Vert A \Vert m >> 1, P \leftarrow P-1$	S_2	3	DD

Tabla 7.7: Señales de control y transferencias de registros asociadas para el multiplicador binario

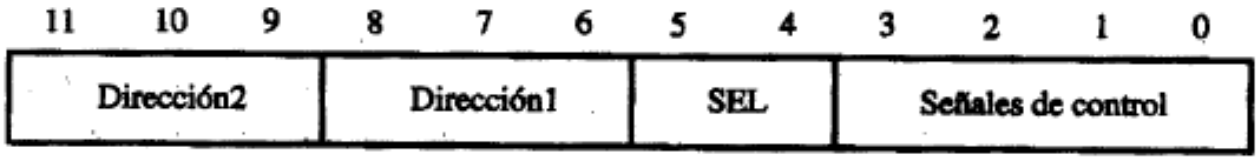


Figura 7.32: Formato de la microinstrucción

SEL		Secuenciamiento de microórdenes
Nombre simbólico	Código binario	
SIG	00	RDC ← Dirección1
DI	01	$\overline{I}$ RDC ← Dirección1 I RDC ← Dirección2
Ds_0	10	$\overline{s}_0$ RDC ← Dirección1 s_0 RDC ← Dirección2
Ds_1	11	$\overline{s}_1$ RDC ← Dirección1 s_1 RDC ← Dirección2

Tabla 7.8: Definición del campo de selección SEL para el secuenciamiento de la unidad de control del multiplicador

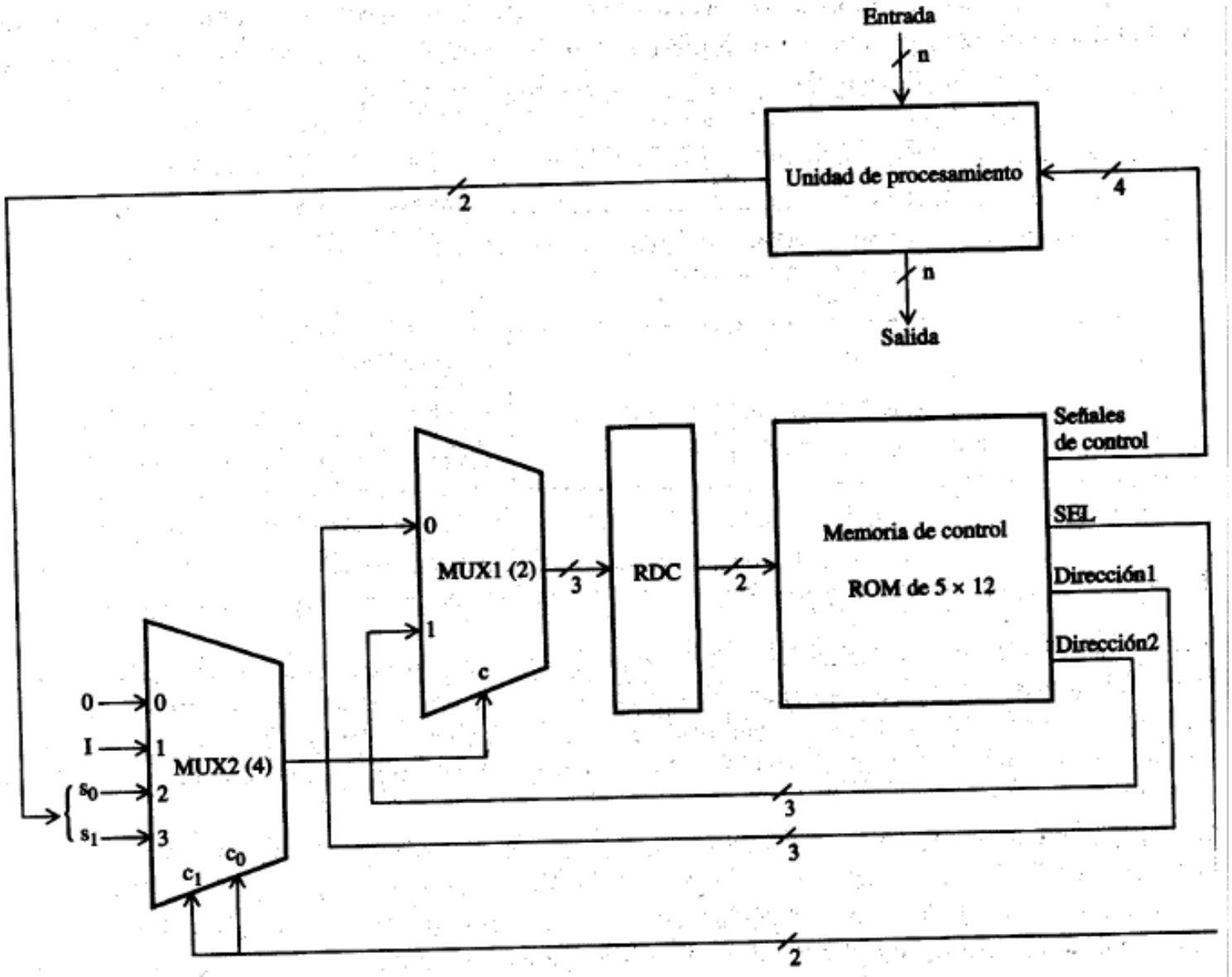


Figura 7.33: Unidad de control microprogramada del multiplicador

Dirección	Sentencias de transferencia simbólica
S_0	I: RDC $\leftarrow S_{00}$, $\overline{I}$: RDC $\leftarrow S_0$
S_{00}	C $\leftarrow$ 0, A $\leftarrow$ 0, P $\leftarrow$ n - 1, RDC $\leftarrow S_1$
S_1	s_0: RDC $\leftarrow S_{10}$, $\overline{s}_0$: RDC $\leftarrow S_2$
S_{10}	A $\leftarrow$ A + M, C $\leftarrow c_a$, RDC $\leftarrow S_2$
S_2	C $\leftarrow$ 0, C || A || m $\leftarrow$ C || A || m >> 1, s_1: RDC $\leftarrow S_0$, $\overline{s}_1$: RDC $\leftarrow S_1$, P $\leftarrow$ P - 1

Tabla 7.9: Microprograma del multiplicador en la notación de transferencia de registro

Dirección	Dirección2	Dirección1	Selección (SEL)	Señales de control
S_0	S_{00}	S_0	DI	Ninguna
S_{00}	---	S_1	SIG	IN, BC
S_1	S_{10}	S_2	Ds_0	Ninguna
S_{10}	---	S_2	SIG	SC
S_2	S_0	S_1	Ds_1	BC, DD

Tabla 7.10: Microprograma simbólico del multiplicador

Dirección	Dirección2	Dirección1	Selección (SEL)	Señales de control
000	001	000	01	0000
001	000	010	00	0101
010	011	100	10	0000
011	000	100	00	0010
100	000	010	11	1100

Tabla 7.11: Código binario del microprograma del multiplicador