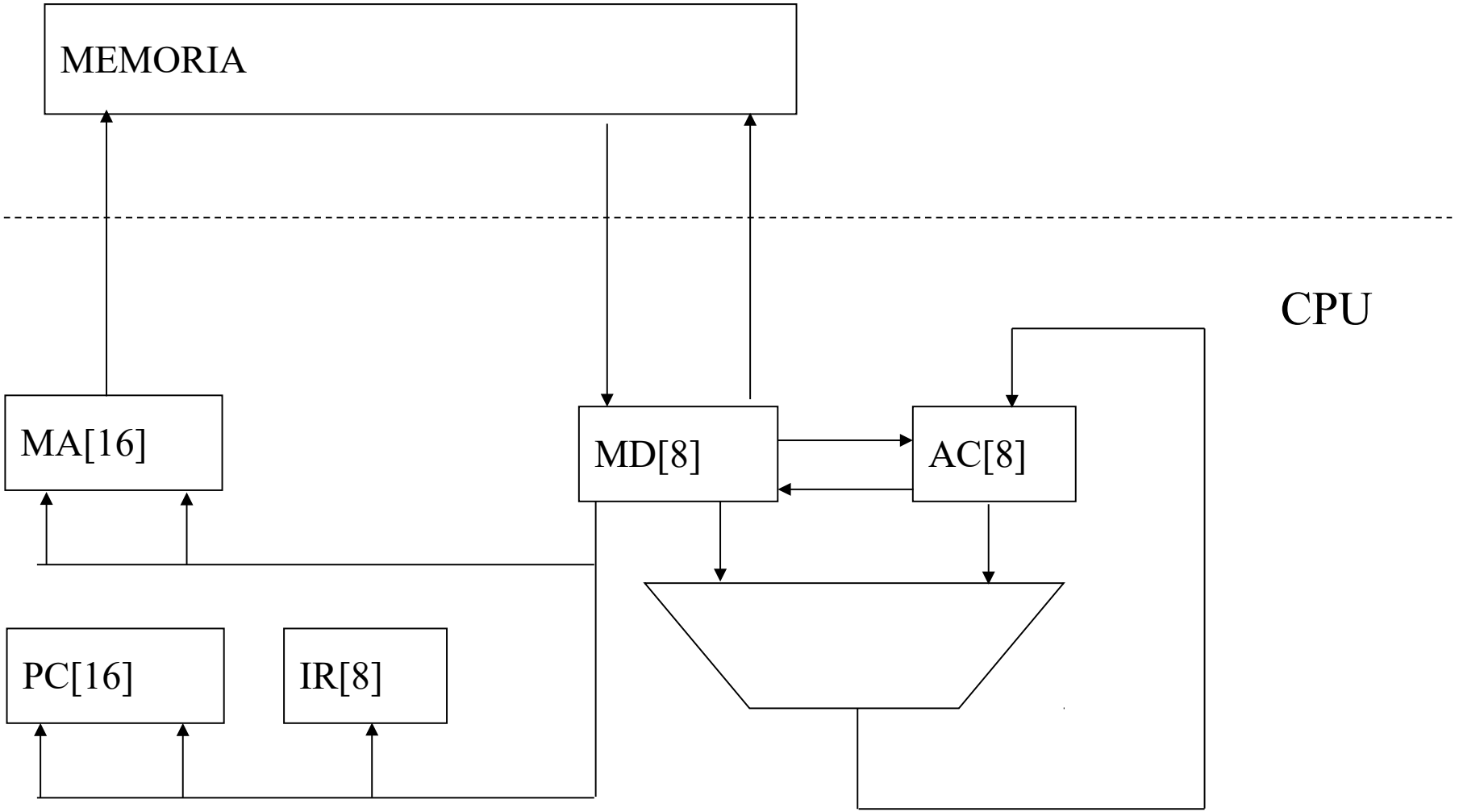
Arquitectura



Arquitectura

- **IR**: Registro de instrucciones. Almacena el código de operación durante la ejecución de la instrucción.
- **AC**: Acumulador. Contiene uno de los operandos y recibe el resultado de todas las operaciones realizadas por la ALU.
- **PC**: Contador de programa. Contiene siempre la dirección de la próxima palabra a leer desde la memoria de programa. Se incrementa cada vez que se lee una palabra. Se modifica además en las instrucciones de salto.
- **MA**: Dirección de memoria. Registro auxiliar, sus salidas son las líneas del bus de direcciones.
- **MD**: Datos de memoria, Registro auxiliar para almacenar temporalmente los datos transferidos desde y hacia memoria. Contiene temporalmente al segundo operando en las operaciones de la ALU.

- Notación: M<MA> memoria M con registro MA conectado a direcciones.

Fases en la ejecución

- Condiciones iniciales:
  - Instrucciones y datos cargados en memoria.
  - PC contiene dirección de comienzo de la próxima instrucción.
  - Los demás registros no inicializados o con lo que quedó de instrucciones anteriores.

Fases en la ejecución

- Fases en la ejecución de una instrucción:
  - Búsqueda del código de operación (opcode fetch).
  - Decodificación.
  - Ejecución.
- Instrucciones CPU ejemplo
  - 1 byte de OPCODE
  - 0, 1 o 2 bytes de operando

Búsqueda del OPCODE

- Secuencia RTL de 3 pasos
  - `1. MA <-- PC`
  - `2. MD <-- M<MA>; PC <-- INC(PC)`
  - `3. IR <-- MD`

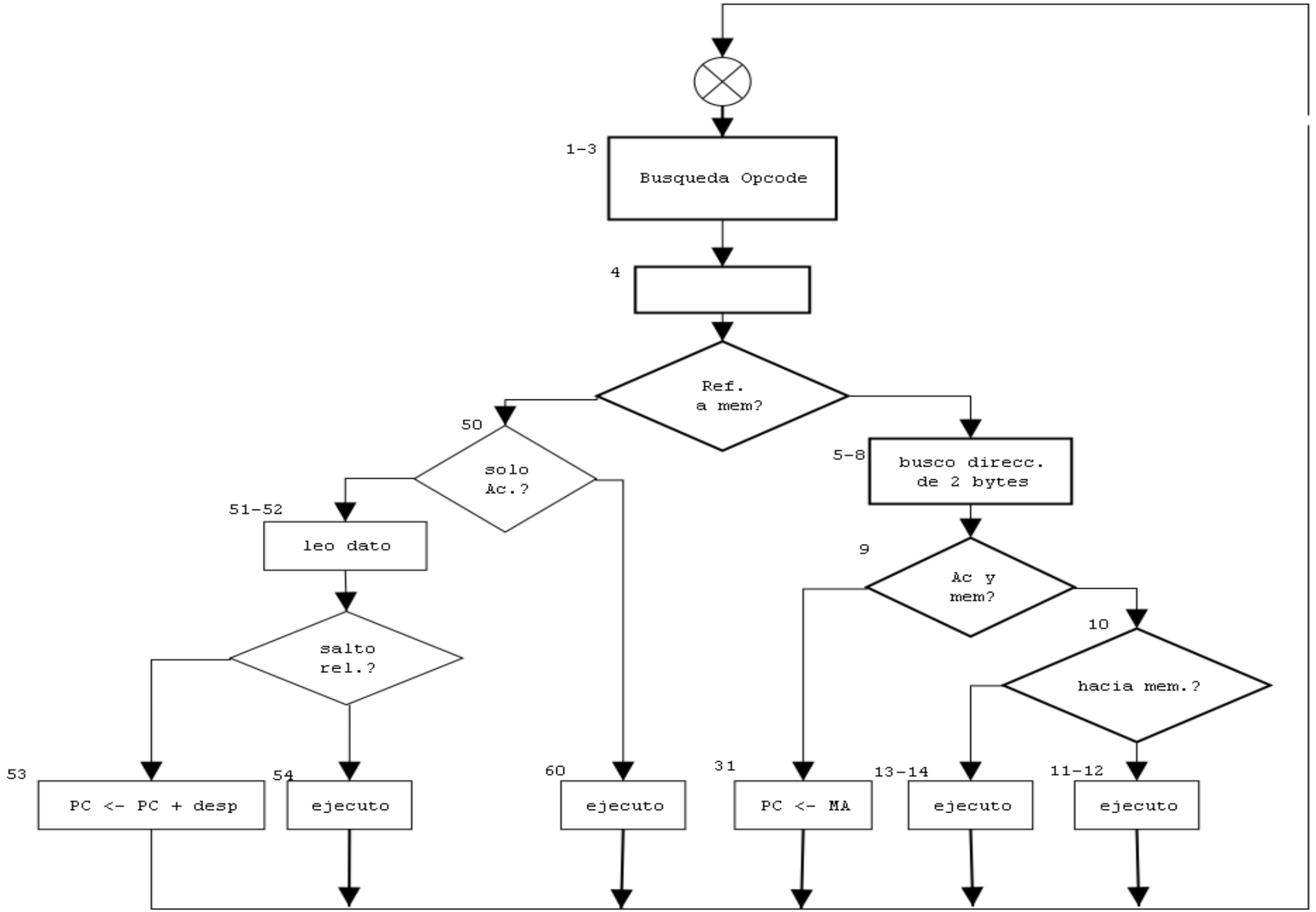
Repertorio de Instrucciones

- Repertorio reducido
- Clasificadas según tareas del bloque de control
  - Instrucciones de un solo operando (acumulador)
  - Instrucciones que requieren solo un byte adicional
    - Saltos relativos
    - Operaciones de acumulador con dato inmediato
  - Instrucciones que requieren leer una dirección de memoria:
    - Operaciones entre memoria y acumulador, direcc. directo
    - Saltos absolutos

Repertorio de Instrucciones

- Codificación

IR[7]	IR[6]	IR[5]	IR[4]	Tipo de instrucción	Ejemplo
0	0	X	X	Solo acumulador	INC A
0	1	0	X	Saltos relativos	JR despl
0	1	1	X	Dato inmediato	ADD A, n
1	0	X	X	Saltos absolutos	JP dir
1	1	X	X	Acumulador y memoria	SUB A, (dir)

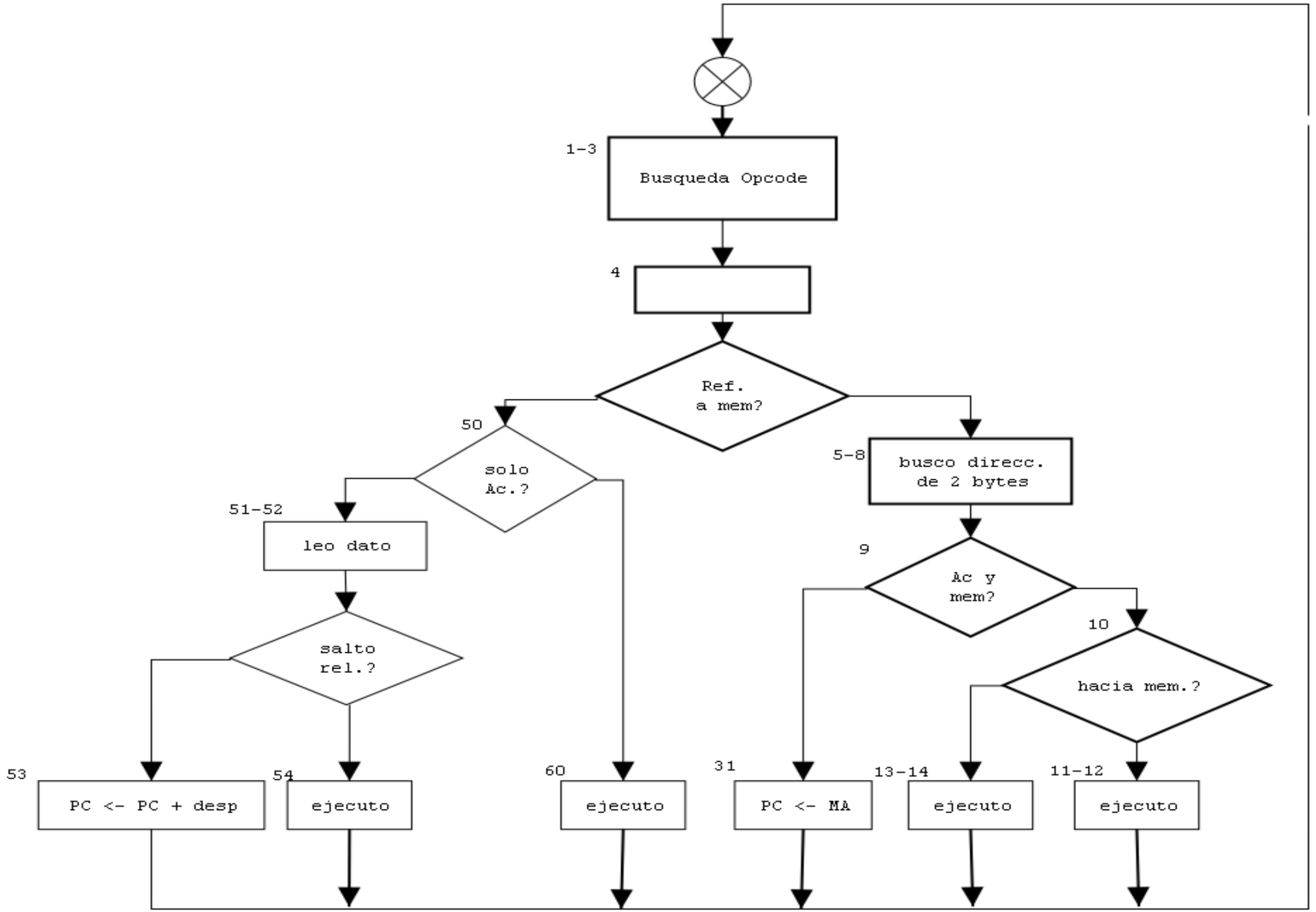

1-3
Busqueda Opcode
4
Ref.
a mem?
50
solo
Ac.?
51-52
leo dato
salto
rel.?
53
PC <- PC + desp
54
ejecuto
60
ejecuto
5-8
busco direcc.
de 2 bytes
9
Ac y
mem?
31
PC <- MA
10
hacia mem.?
13-14
ejecuto
11-12
ejecuto

Secuencia RTL

```
   4. -> ( IR[7] = 0 ) / 50 - solo ac, jr, dato inmediato
         -- instrucciones que leen una dirección
            --    MA ← M<NN+1>, M<NN>
            --    PC ← PC + 2
            --       donde NN es la dir del opcode
   5. MA <-- PC; PC <-- INC(PC)
-  6. MD <-- M<MA>; MA <-- PC; PC <-- INC(PC)
   7. MD <-- M<MA>; MA[7..0] <-- MD
   8. MA[15..8] <-- MD
   9. -> (IR[6] = 0) / (31) -- si salto abs., a paso 31

   10. -> (hacia memoria) / (13)  -- guardar ac en memoria
```

Secuencia RTL

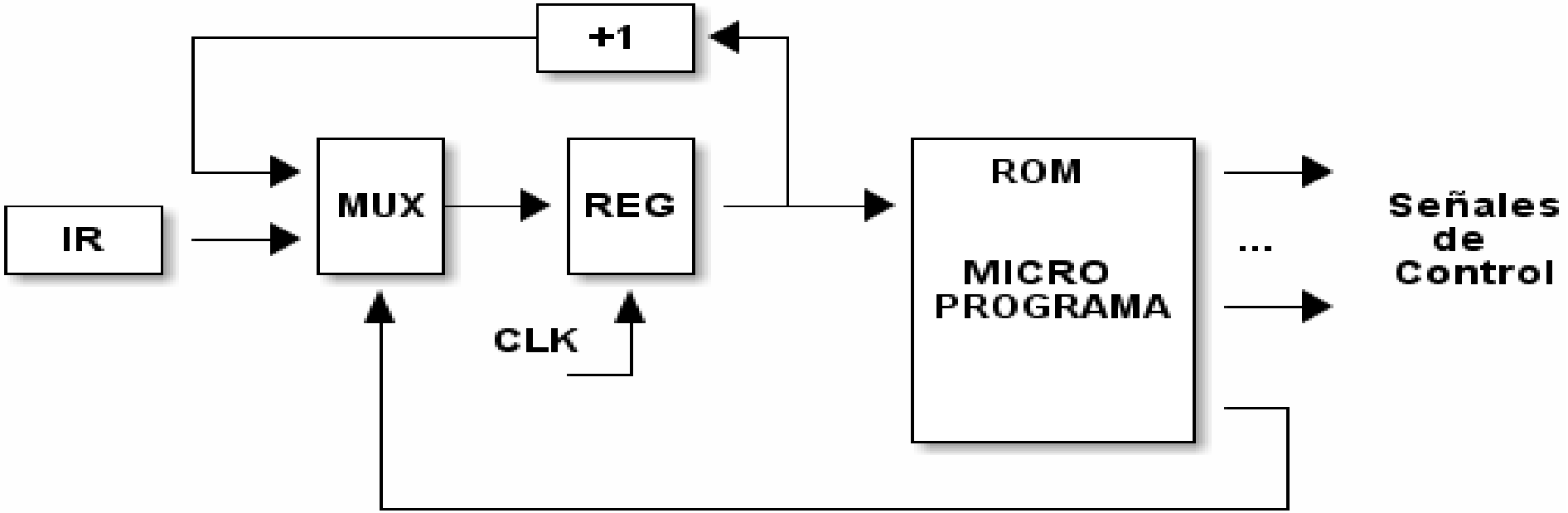
```
--------- operación entre acum. y memoria
11. MD < M<MA>
12. AC <-- AC op MD  -- op según indique IR[5..0]
    → (1)

--------- transferencia hacia memoria
13. MD <-- AC
14. M<MA> <-- MD
  → (1)

----------  Salto absoluto
31. PC  <-- MA
  → (1)
```


1-3
Busqueda Opcode
4
Ref.
a mem?
50
solo
Ac.?
51-52
leo dato
salto
rel.?
53
PC <- PC + desp
54
ejecuto
60
ejecuto
5-8
busco direcc.
de 2 bytes
9
Ac y
mem?
31
PC <- MA
10
hacia mem.?
13-14
ejecuto
11-12
ejecuto

Secuencia RTL

```
------------ Si solo sobre el acumulador salto a paso 60
50. → ( IR[6] = 0 ) / (60)

----------- Instrucciones que leen un byte (jr o dato inmediato)
51. MA ← PC; PC ← INC(PC)
52. MD ← M<MA>; → (carga dato inmediato) / (55)

53. PC ← PC + MD; → (1)      -- salto relativo

------------ carga de dato inmediato
55. AC ← AC op MD; → (1)     -- op según IR[ 5..0 ]

-----------  Instrucciones que operan solo sobre el acumulador
60. AC ← f(AC, IR); → (1)
```

Solución alternativa: Memoria de microprograma



Solución alternativa: Memoria de microprograma

- Alternativa para diseñar el bloque de control
- Memoria de “microprograma”
  - Salidas de la memoria son las señales de control
  - Direcciones:
    - Inicializada desde registro IR al comienzo de cada instrucción
    - Incrementada en cada flanco de reloj
  - Para comenzar una nueva instrucción una de las salidas:
    - Controla la carga carga de la dirección desde el registro IR