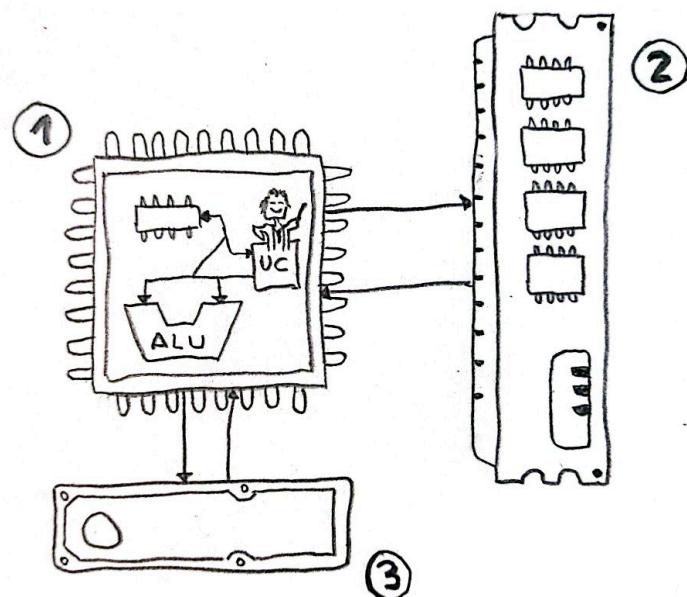


# MÁQUINA DE VON NEUMANN



Práctica 01  
Entornos de Desarrollo (Bartomeu Segura i Duran)

---

---

Adrián Cervera Fernández

<b>1. INTRODUCCIÓN.....</b>	<b>2</b>
<b>2. MÁQUINA DE VON NEUMANN.....</b>	<b>2</b>
2.1. Definición de la máquina de Von Neumann.....	2
2.2. Componentes de la máquina de Von Neumann.....	2
2.2.1. Unidad central de proceso.....	2
2.2.2. Memoria.....	2
2.2.3. Unidad de control.....	3
2.2.4. Unidad Aritmética-lógica.....	4
2.2.4 Buses (de datos, de control y de direcciones).....	4
2.2.4.1. Bus de datos.....	4
2.2.4.2. Bus de direcciones.....	4
2.2.4.3. Bus de control.....	5
2.3. Arquitectura de Von Neumann.....	5
<b>3. EJERCICIO PRÁCTICO RESUELTO.....</b>	<b>6</b>

## 1. INTRODUCCIÓN

En el presente estudio se persigue adquirir conocimientos acerca del funcionamiento de la máquina de Von Neumann, incluyendo el procesamiento de instrucciones y datos.

Se explicará el funcionamiento de la arquitectura de Von Neumann, y a continuación se simulará una operación con un desarrollo básica a partir de la máquina de Von Neumann.

## 2. MÁQUINA DE VON NEUMANN

### 2.1. Definición de la máquina de Von Neumann

---

Se conoce como máquina de Von Neumann al diseño teórico que define como debe funcionar un sistema informático y sus componentes, y a su vez la relación que tienen entre sí para realizar las operaciones.

### 2.2. Componentes de la máquina de Von Neumann

---

Para entender como funciona la máquina de Von Neumann, hay que entender los principales componentes de la misma, los cuales son:

#### 2.2.1. Unidad central de proceso

---

Unidad Central de Proceso; más conocida como CPU, es el componente encargado de ejecutar las instrucciones de un programa, a través de transistores, con los que procesa los datos y controla la máquina.

#### 2.2.2. Memoria

---

La memoria, conocida como memoria RAM, se encarga de almacenar todas las instrucciones y los datos, lo que hace que tengan que tener un orden secuencial, ya que no puede acceder a la vez a las instrucciones y los datos porque están guardados en la misma memoria.

Cada posición de memoria tiene asignada una dirección única, donde la CPU gestionará el acceso a la dirección adecuada.

La memoria se basa en dos columnas, en una se encuentra el registro de direcciones y en la otra el registro de contenido.

Esta memoria utiliza los números binarios para su funcionamiento, ya que los bits están diseñados para leer 1 y 0.

**TABLA DE MEMORIA**

DIRECCIÓN	CONTENIDO
0000	00000100
0001	00001111
0010	00100010
0011	00010011
0100	01110011
0101	00000111

En el registro de direcciones, se guardan las direcciones donde se guardará o leerá el dato, y en el registro de contenido, se guardará el dato leído o el que se guardará.

### 2.2.3. Unidad de control

La unidad de control es un componente vital de la CPU para llevar a cabo la arquitectura de Von Neumann. Se encarga de controlar todas las operaciones que realiza la CPU, coordinando las instrucciones para que se ejecuten en el orden correcto.

Como su propio nombre indica, la unidad de control actúa dirigiendo todas las acciones en el sistema, pero no realiza ningún cálculo.

La unidad de control recibe las instrucciones de la memoria, y envía las señales necesarias a los diferentes componentes para que procesen o almacenen los datos según sea necesario.



Así es como funciona gráficamente la unidad de control.

### 2.2.4. Unidad Aritmética-lógica

La unidad aritmético-lógica es la encargada de todas las operaciones, tanto lógicas (comparaciones) como matemáticas (aritméticas; sumas, multiplicaciones...), que se necesitan para ejecutar las instrucciones de un programa.

Esta unidad recibe los datos que se tienen que procesar de la memoria, y recibe las órdenes de las operaciones que tiene que realizar de la unidad de control.



Aquí podemos ver la explicación gráfica de la unidad aritmético-lógica, que interviene en la fase de ejecución de las instrucciones, realizando las operaciones necesarias para su ejecución.

### 2.2.4 Buses (de datos, de control y de direcciones)

En la arquitectura de Von Neumann, los buses, permiten a todos los elementos intercambiar los datos o las instrucciones para manejar la máquina, entre ellos. Hay 3 tipos de buses, y cada uno tiene unas características particulares en el funcionamiento del sistema.

#### 2.2.4.1. Bus de datos

El bus de datos es el canal por donde se mueven los datos entre los distintos elementos de la máquina.

Por ejemplo, si la CPU tiene que restar dos números, el bus de datos llevará los dos números desde la memoria hasta la unidad aritmético-lógica, dónde se realizará la operación.

#### 2.2.4.2. Bus de direcciones

Este bus transmite las direcciones de la memoria que serán utilizadas por la CPU para utilizar los datos que necesite.

La característica principal de este bus es que es unidireccional, ya que solo necesita transmitir la información en una sola dirección; como de la CPU a la memoria.

### 2.2.4.3. Bus de control

El bus de control se encarga de gestionar y coordinar las operaciones del sistema, es decir, transporta todas las órdenes generadas por la CPU para controlar los diferentes procesos de la máquina.

Este es un bus bidireccional, ya que cuando la CPU envía las órdenes a los otros componentes, estos pueden enviar señales de respuesta o de interrupción de vuelta a la CPU.

Este es un ejemplo esquemático de como funcionan los tres buses a la hora de realizar una operación, como por ejemplo una suma.

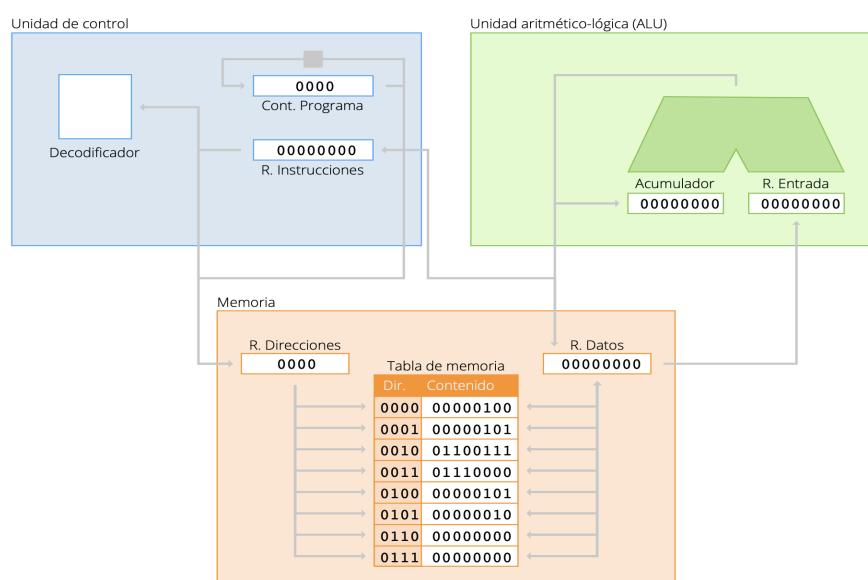


## 2.3. Arquitectura de Von Neumann

La arquitectura de Von Neumann se diseñó con el objetivo de estructurar de manera lógica una máquina, principalmente un ordenador.

Se basaba en tres componentes principales, la memoria, la unidad de control y la unidad aritmético-lógica.

Estos componentes se comunican entre sí para realizar las operaciones.



Esta es la estructura básica de la máquina de Von Neumann.

La estructura de Von Neumann se ha extendido en el tiempo, desde que se inventase en 1945 por John von Neumann, es la base de distintas máquinas, principalmente de los ordenadores.

Una de las características que hizo destacar esta arquitectura fue su memoria unificada, ya que tanto los datos como las instrucciones se guardan en una sola memoria. Esto ha hecho también que la arquitectura de Von Neumann sea usada cada vez menos, porque esta memoria provoca un cuello de botella, puesto que solo se puede acceder a un dato u instrucción a la vez.

Otra de las características es la secuencia que sigue, donde el procesador ejecuta las instrucciones una tras otra.

Las principales ventajas de la arquitectura de Von Neumann son la simplicidad de su estructura y la variedad para ejecutar programas con un mismo diseño, simplemente cambiando el hardware.

Un ejemplo de ello sería que se ha usado la misma estructura tanto para hacer un ordenador que para hacer una lavadora.

### 3. EJERCICIO PRÁCTICO RESUELTO

Para explicar como es la arquitectura de la máquina de Von Neumann, a continuación se expondrá un ejemplo práctico de una **suma sencilla (5+2)** mediante un programa en lenguaje de bajo nivel.

El procedimiento para llevar a cabo la operación será el siguiente:

- 1 - La unidad de control envía una micro-orden para llevar el contenido desde el contador de programa hasta el registro de direcciones.
- 2 - El contador del programa aumenta en 1, así que su contenido será la próxima dirección que se ejecutará.
- 3 - Se selecciona la posición de la memoria que indica el registro de direcciones y se hace una lectura en la memoria.
- 4 - La instrucción que se tiene que ejecutar se deposita en el registro de datos.
- 5 - Se traslada la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.
- 6 - El decodificador interpreta la instrucción que serán los primeros 4 bits, que es el código de la operación.

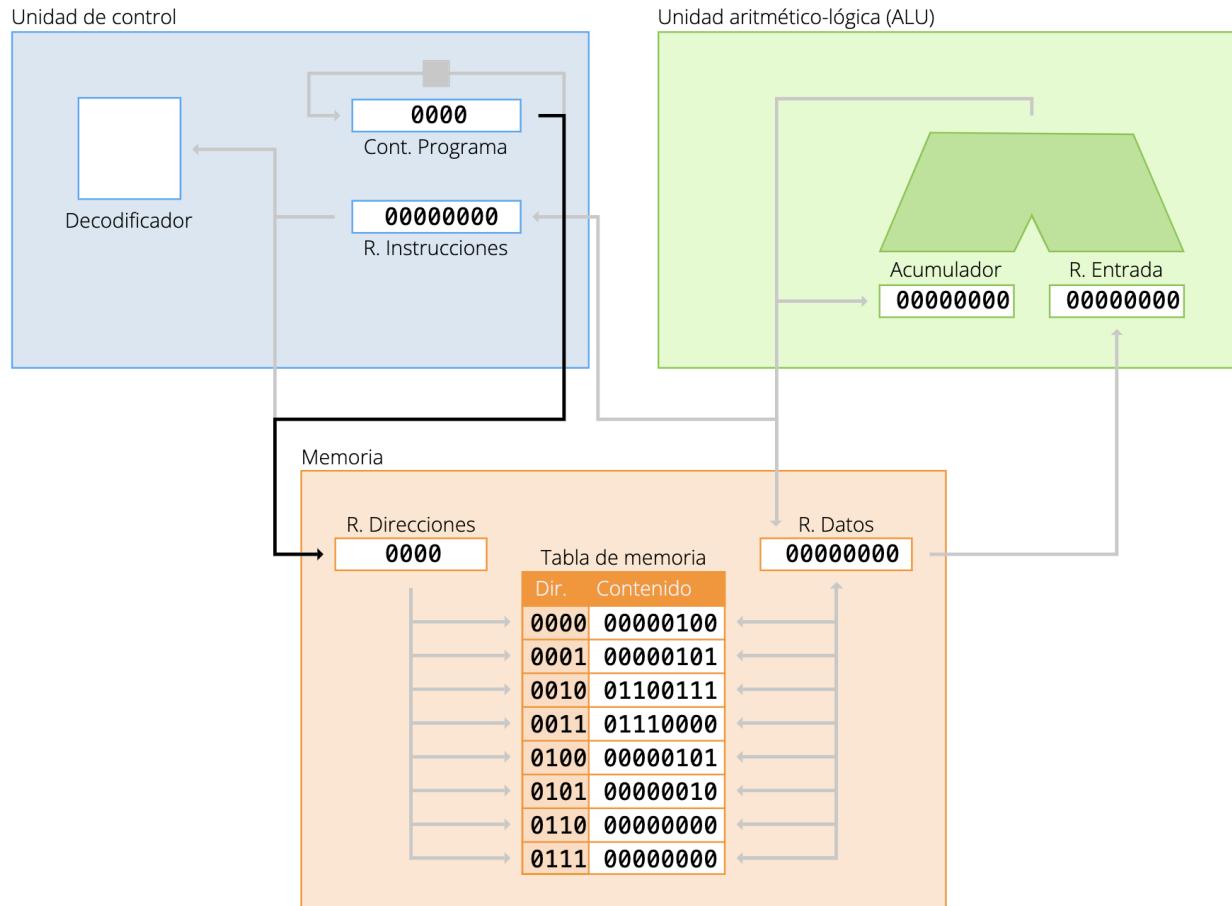
- 7 - El registro de instrucciones envía los últimos 4 bits al registro de direcciones.
- 8 - El registro de direcciones busca en la memoria la celda correspondiente y leer el dato.
- 9 - La información se envía al registro de datos.
- 10 - El registro de datos envía la información al registro de entrada.
- 11 - El circuito operacional realiza la operación con el registro acumulador y el registro de entrada, y lo almacena de nuevo en el registro acumulador.
- 12 - La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.
- 13 - El contador de programa aumenta en 1, por lo que su contenido será la próxima instrucción a ejecutar.
- 14 - Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.
- 15 - Se deposita en el registro de datos la acción a ejecutar.
- 16 - Se realiza el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.
- 17 - El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, lo cual quiere decir que interpreta el código de operación.
- 18 - El registro de instrucciones envía los últimos cuatro bits al registro de direcciones.
- 19 - El registro de direcciones busca en la memoria la celda correspondiente y lee el dato.
- 20 - La información es enviada al registro de datos.
- 21 - El registro de datos envía la información al registro de entrada.
- 22 - El circuito operacional realiza la acción con el registro acumulador y el registro de entrada, y lo almacena otra vez en el registro acumulador.
- 23 - La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.
- 24 - El contador de programas aumenta en uno, por lo que su contenido será la dirección de la próxima instrucción a ejecutar.
- 25 - Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.
- 26 - Se deposita en el registro de datos la instrucción a ejecutar.

- 27 - Se realizará el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.
- 28 - El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, es decir, que interpreta el código de la operación.
- 29 - El registro de instrucciones envía los últimos cuatro bits al registro de direcciones.
- 30 - El registro de direcciones busca en la memoria en la celda en la que será guardada el resultado.
- 31 - El registro acumulador envía la información al registro de datos.
- 32 - El registro de datos procede a la escritura de la información en la celda seleccionada por el registro de direcciones.
- 33 - La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.
- 34 - El contador de programa aumenta en uno, por lo que su contenido será la dirección de la próxima instrucción a ejecutar.
- 35 - Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.
- 36 - Se deposita en el registro de datos la instrucción a ejecutar.
- 37 - Se realiza el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.
- 38 - El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, es decir, que interpreta el código de operación.
- 39 - El decodificador interpreta que se finaliza el programa y se para la ejecución.

## SUMAR 5+2

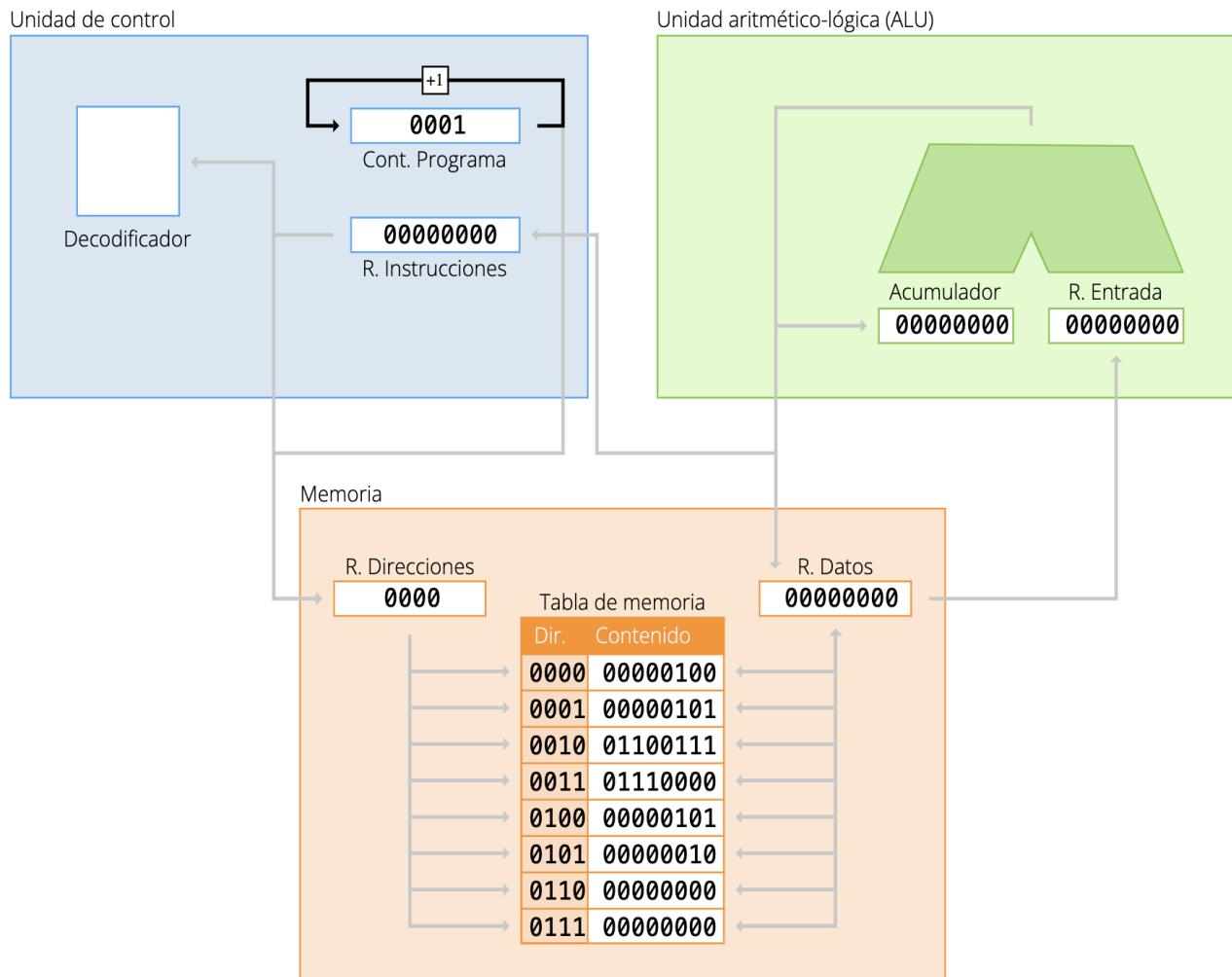
### Ejemplo de la acción número 1.

- La unidad de control envía una micro-orden para llevar el contenido desde el contador de programa hasta el registro de direcciones.



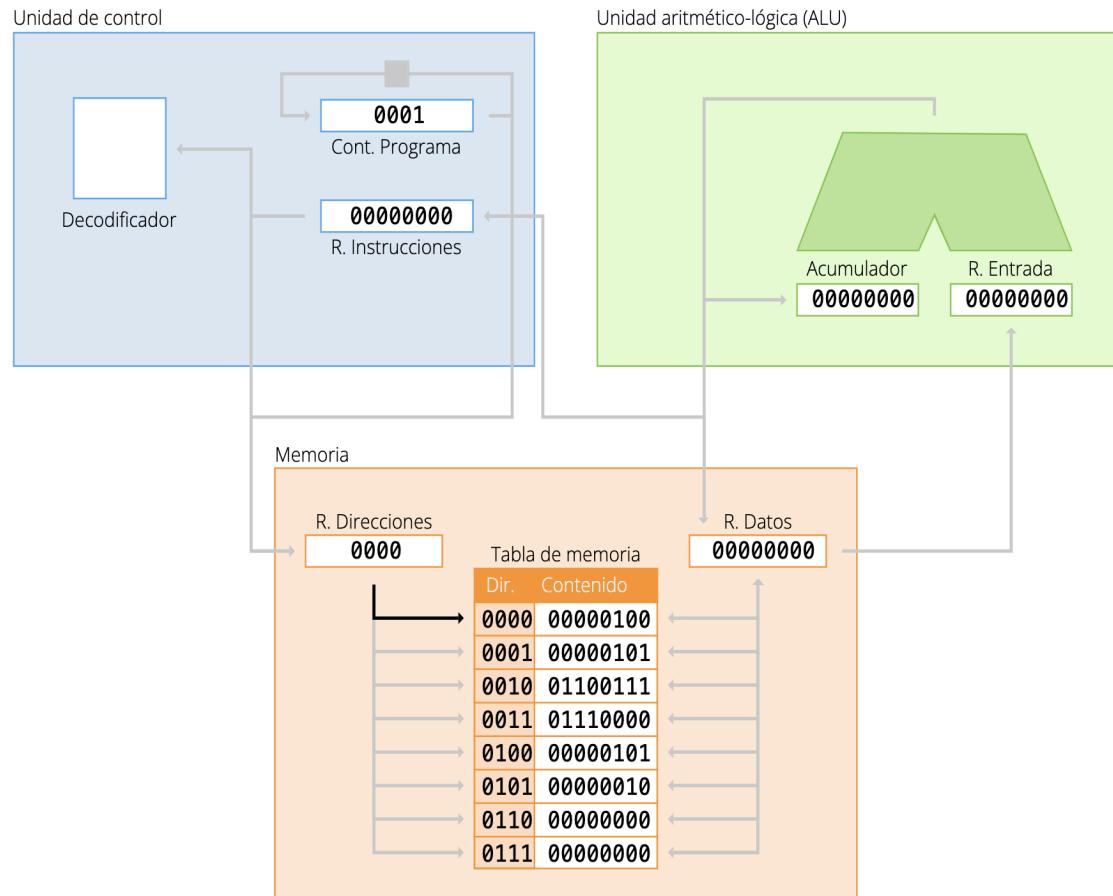
## Ejemplo de la acción número 2

2. El contador del programa aumenta en 1, así que su contenido será la próxima dirección que se ejecutará.



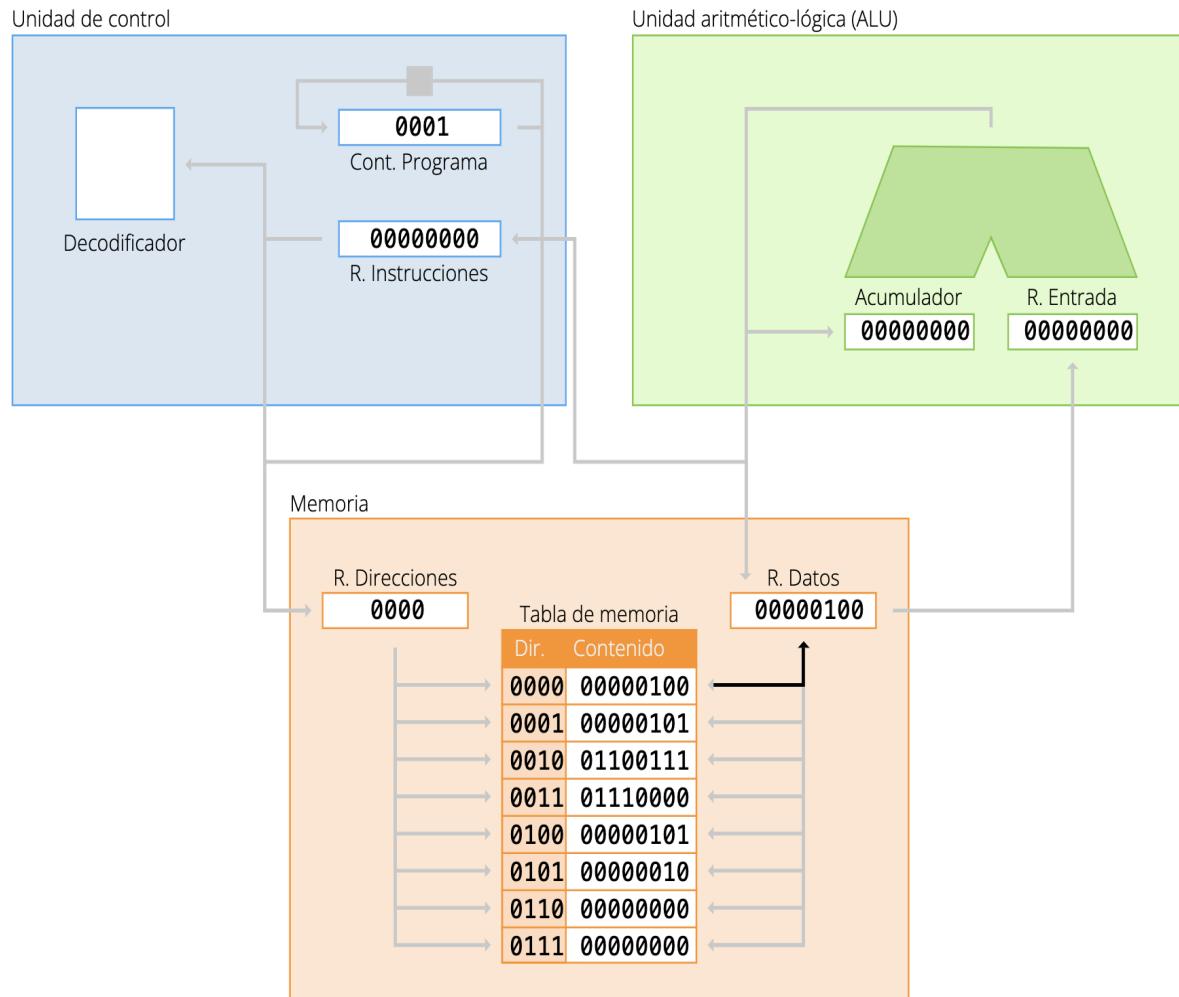
### Ejemplo de la acción número 3.

3. Se selecciona la posición de la memoria que indica el registro de direcciones y se hace una lectura en la memoria.



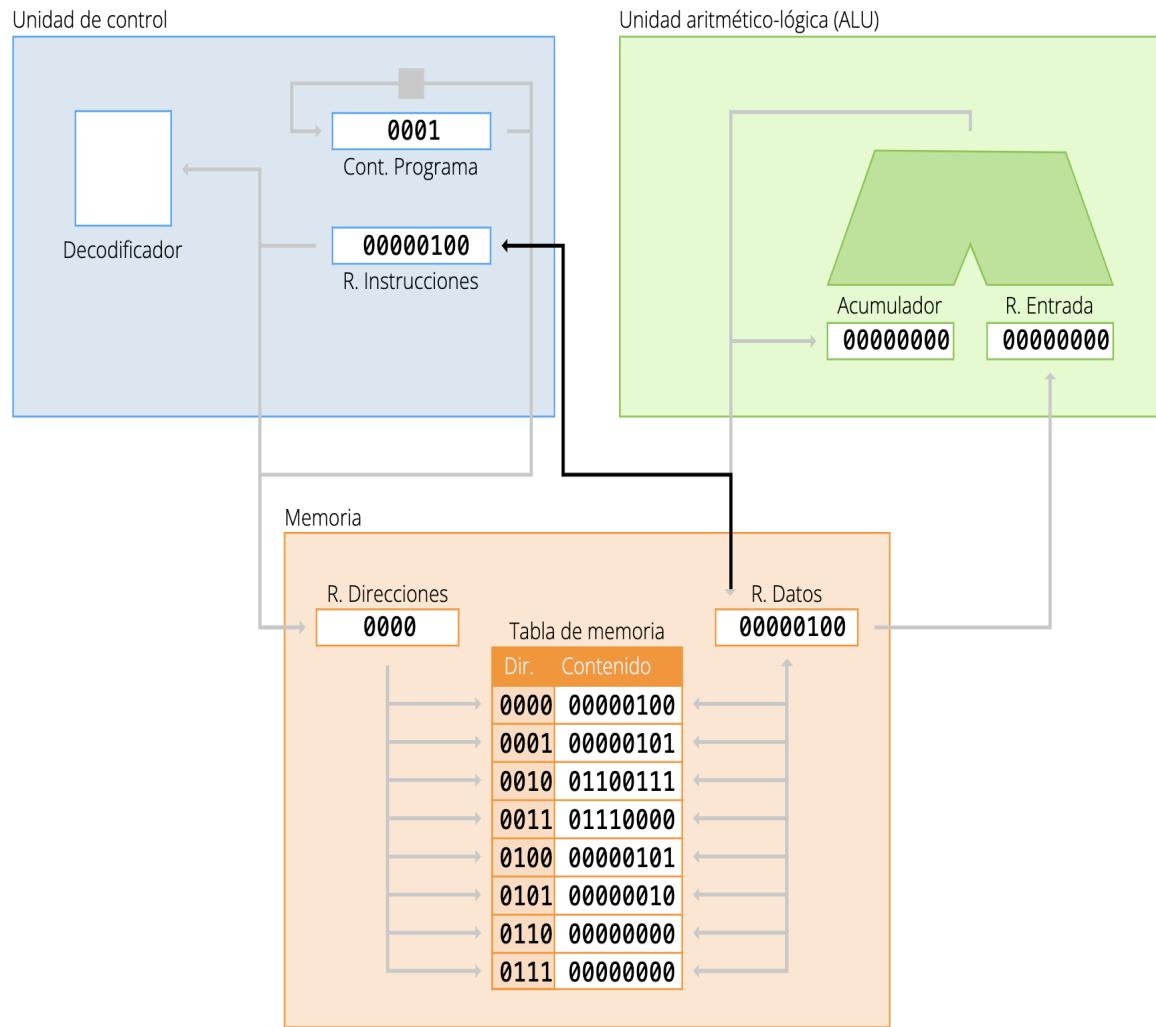
### Ejemplo de la acción número 4.

4. La instrucción que se tiene que ejecutar se deposita en el registro de datos.



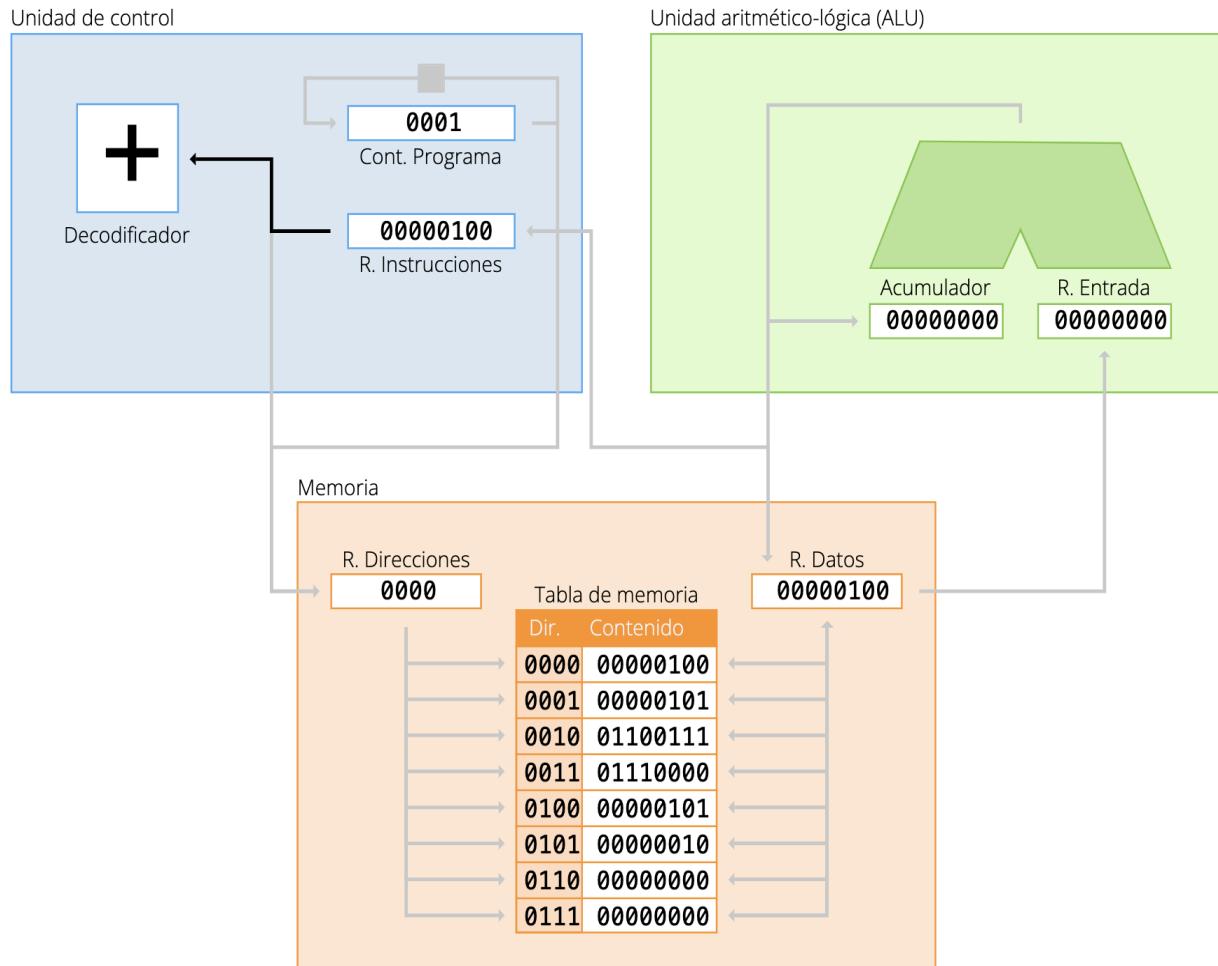
### Ejemplo de la acción número 5.

5. Se traslada la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.



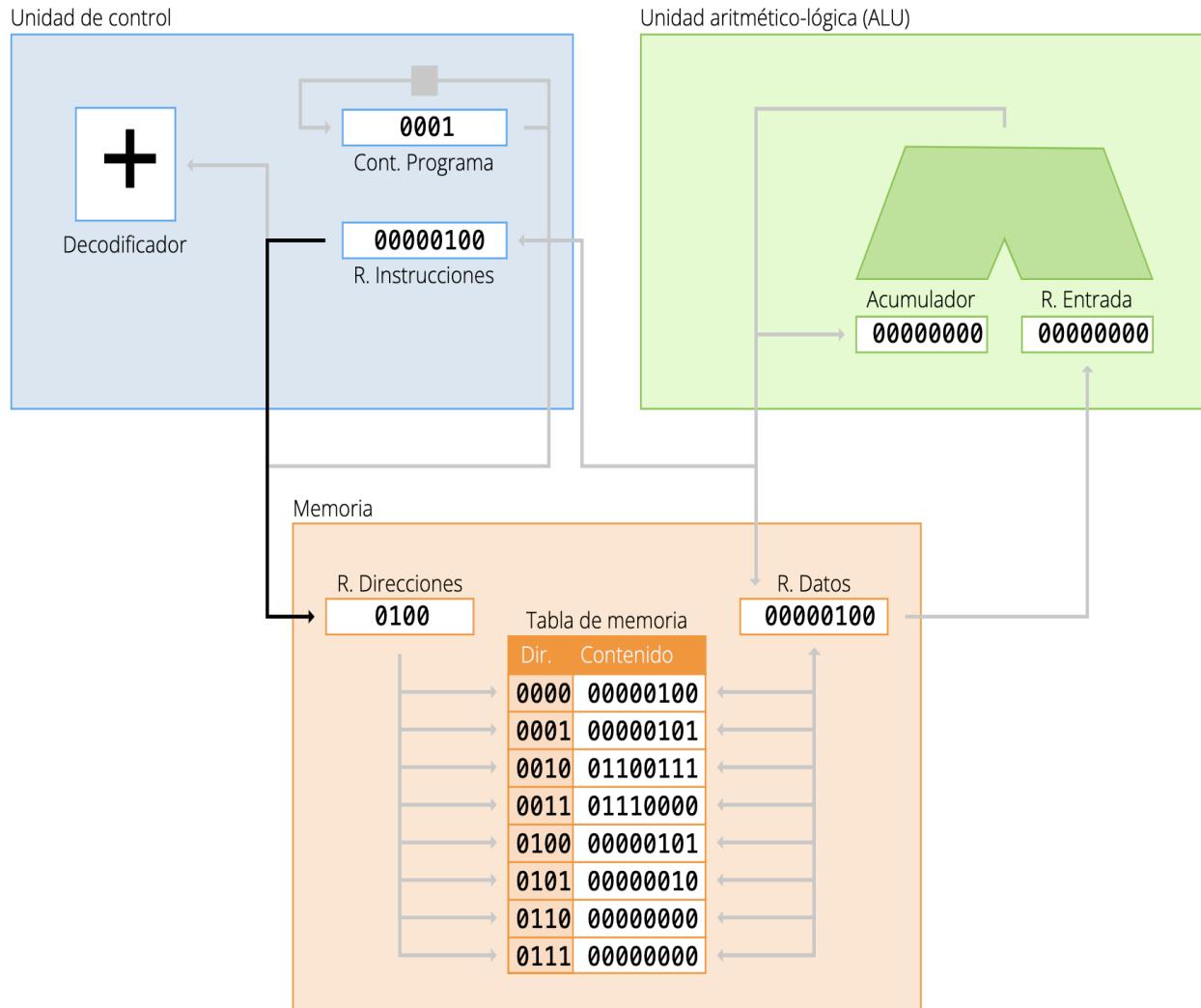
### Ejemplo de la acción número 6.

6. El decodificador interpreta la instrucción que serán los primeros 4 bits, que es el código de la operación.



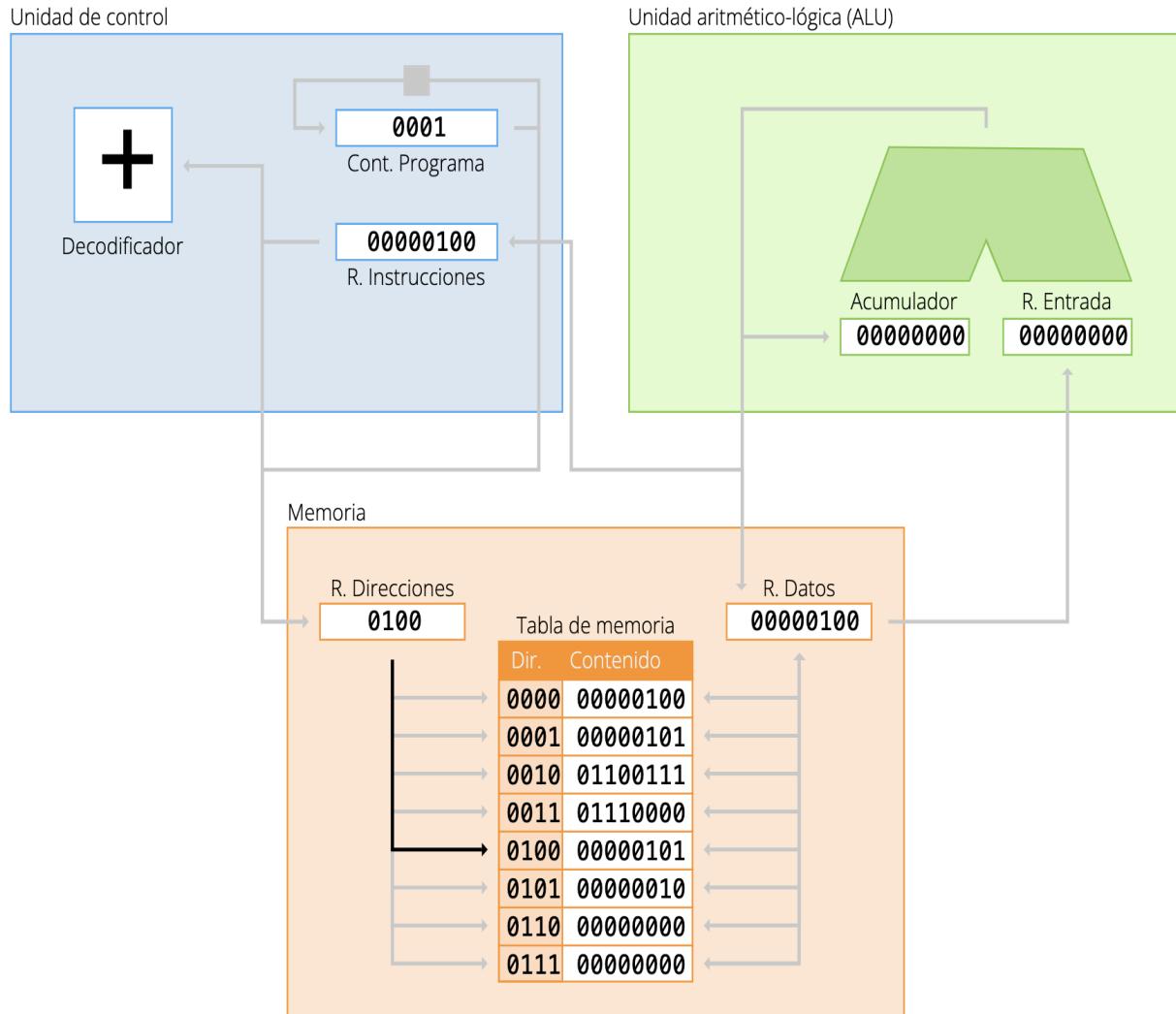
## Ejemplo de la acción número 7.

7. El registro de instrucciones envía los últimos 4 bits al registro de direcciones.



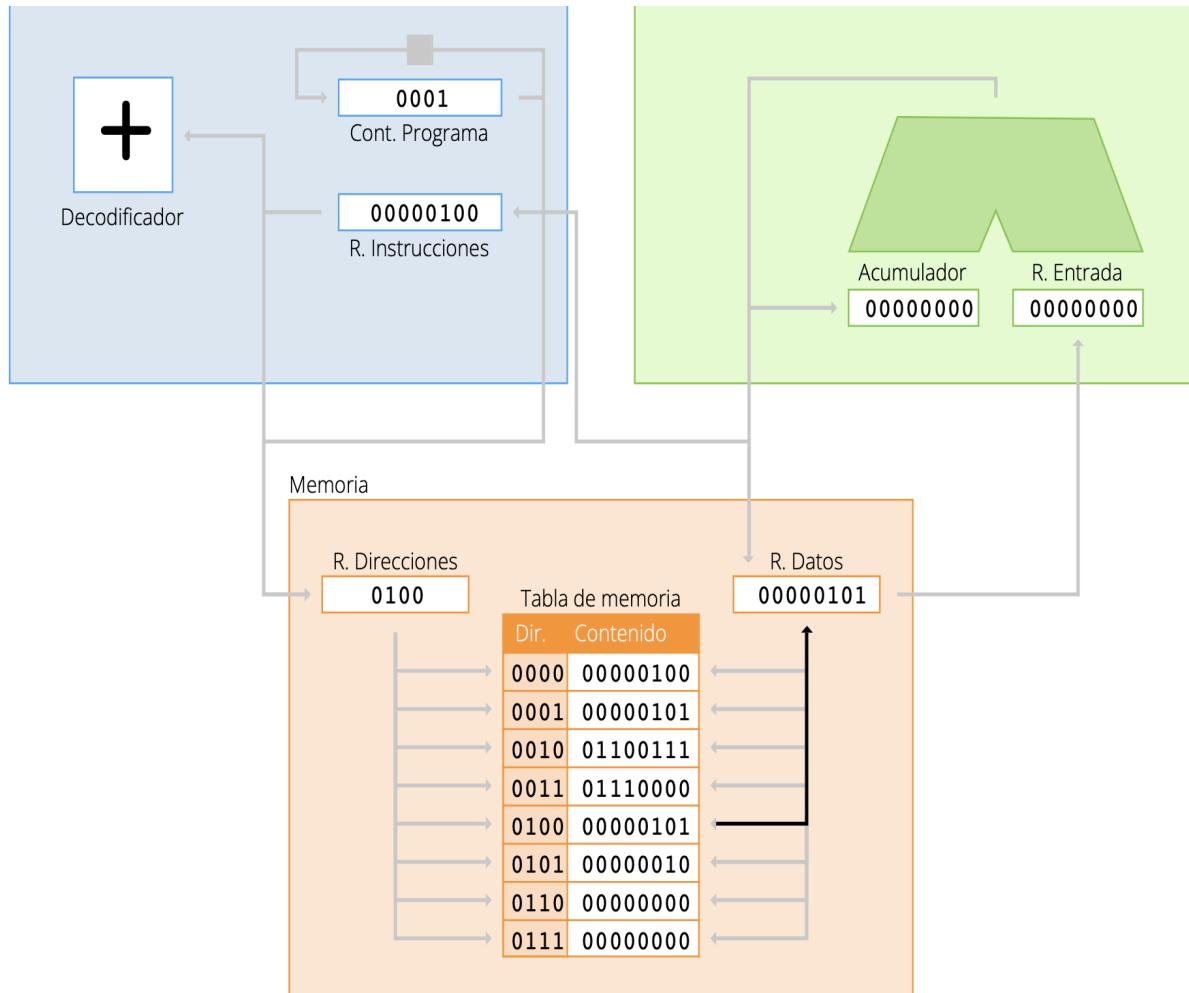
### Ejemplo de la acción número 8.

8. El registro de direcciones busca en la memoria la celda correspondiente y leer el dato.



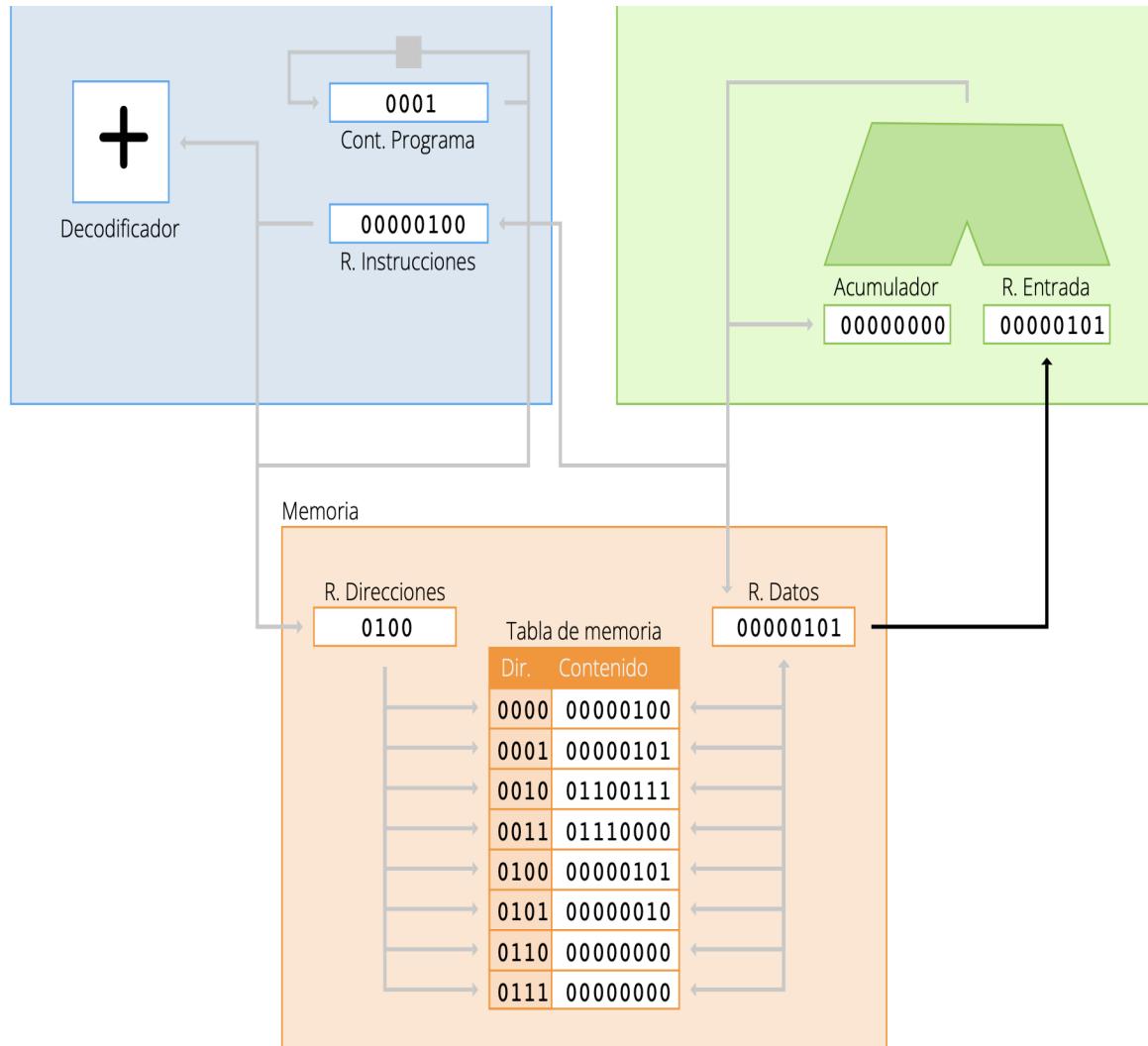
## Ejemplo de la acción número 9.

9. La información se envía al registro de datos.



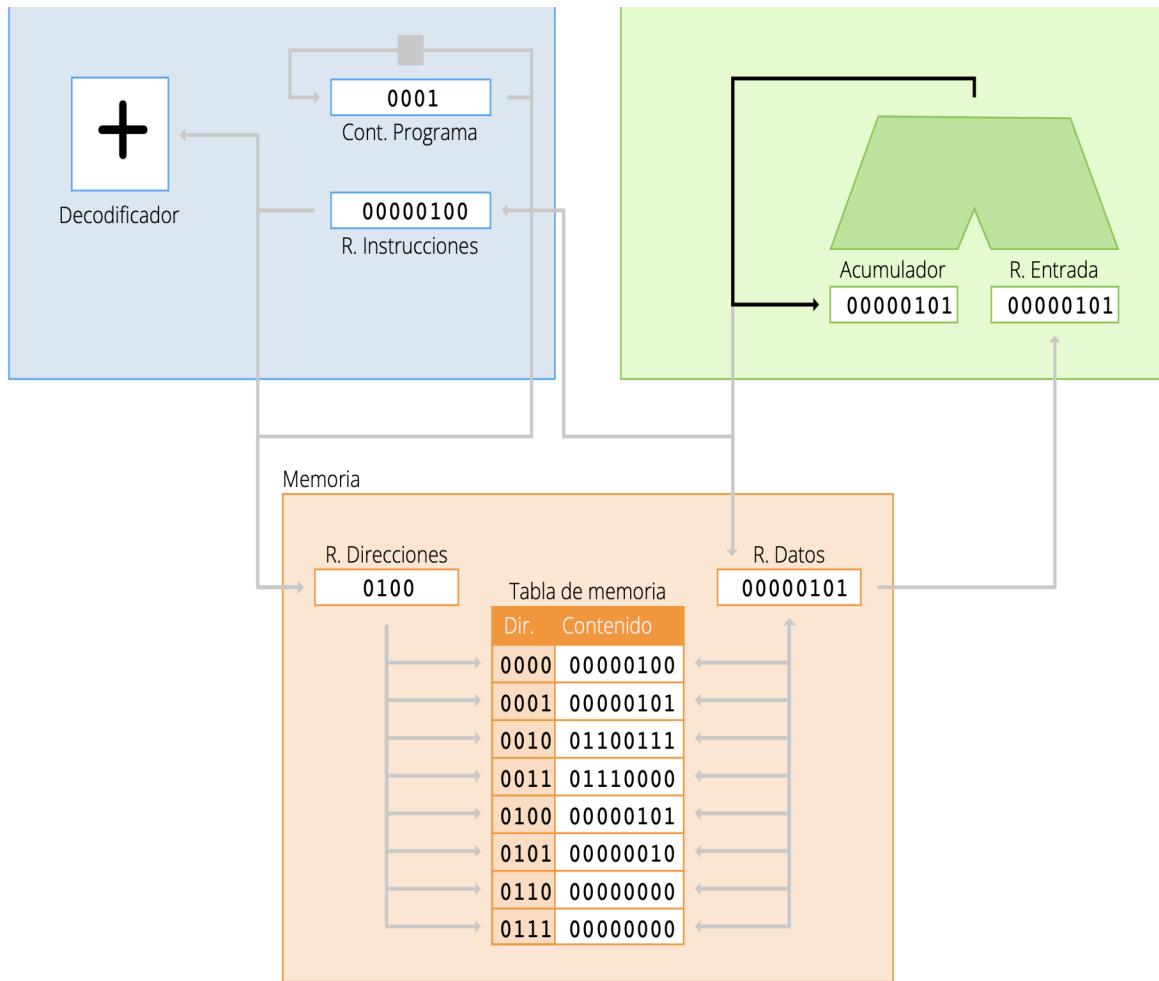
### Ejemplo de la acción número 10.

10. El registro de datos envía la información al registro de entrada.



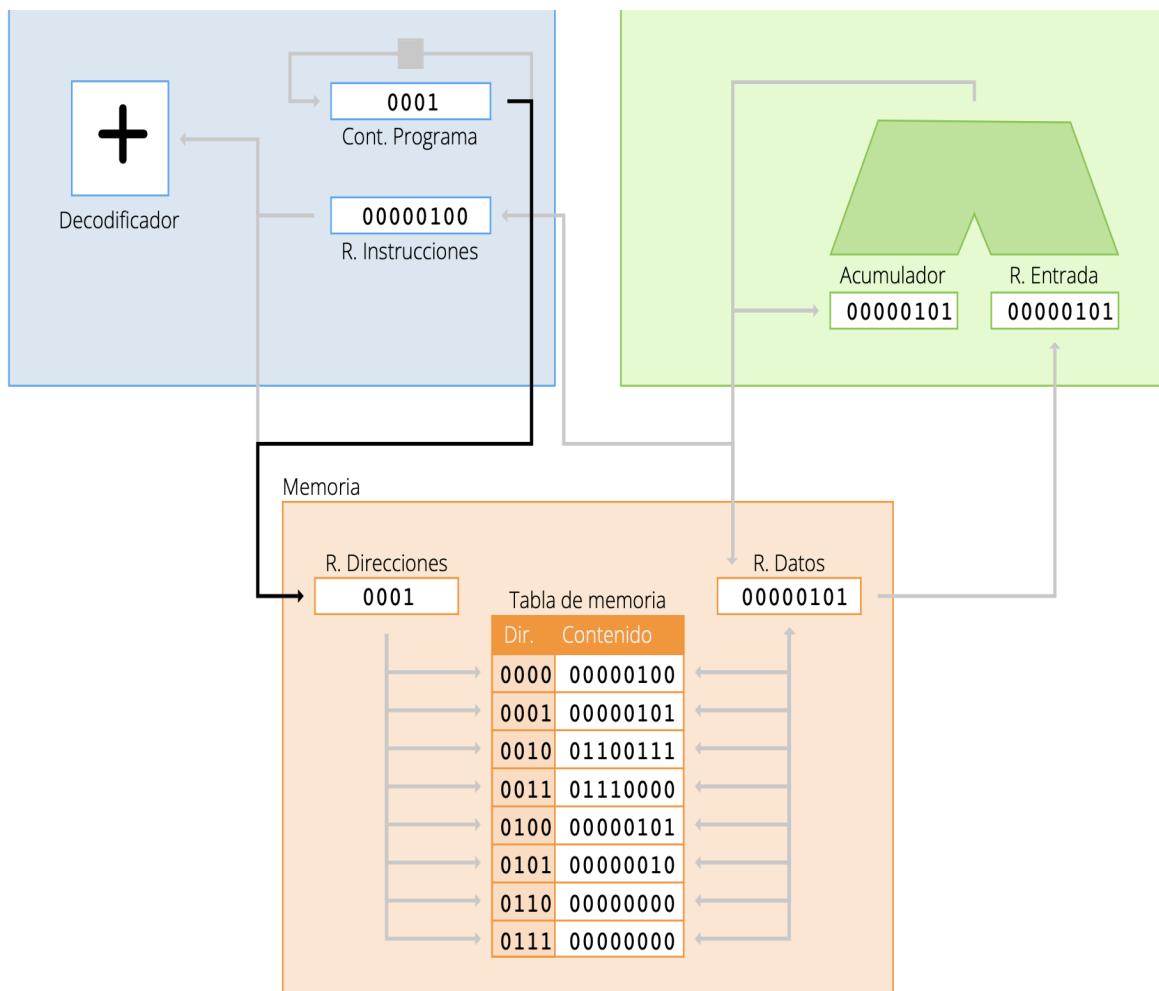
### Ejemplo de la acción número 11.

11. El circuito operacional realiza la operación con el registro acumulador y el registro de entrada, y lo almacena de nuevo en el registro acumulador.



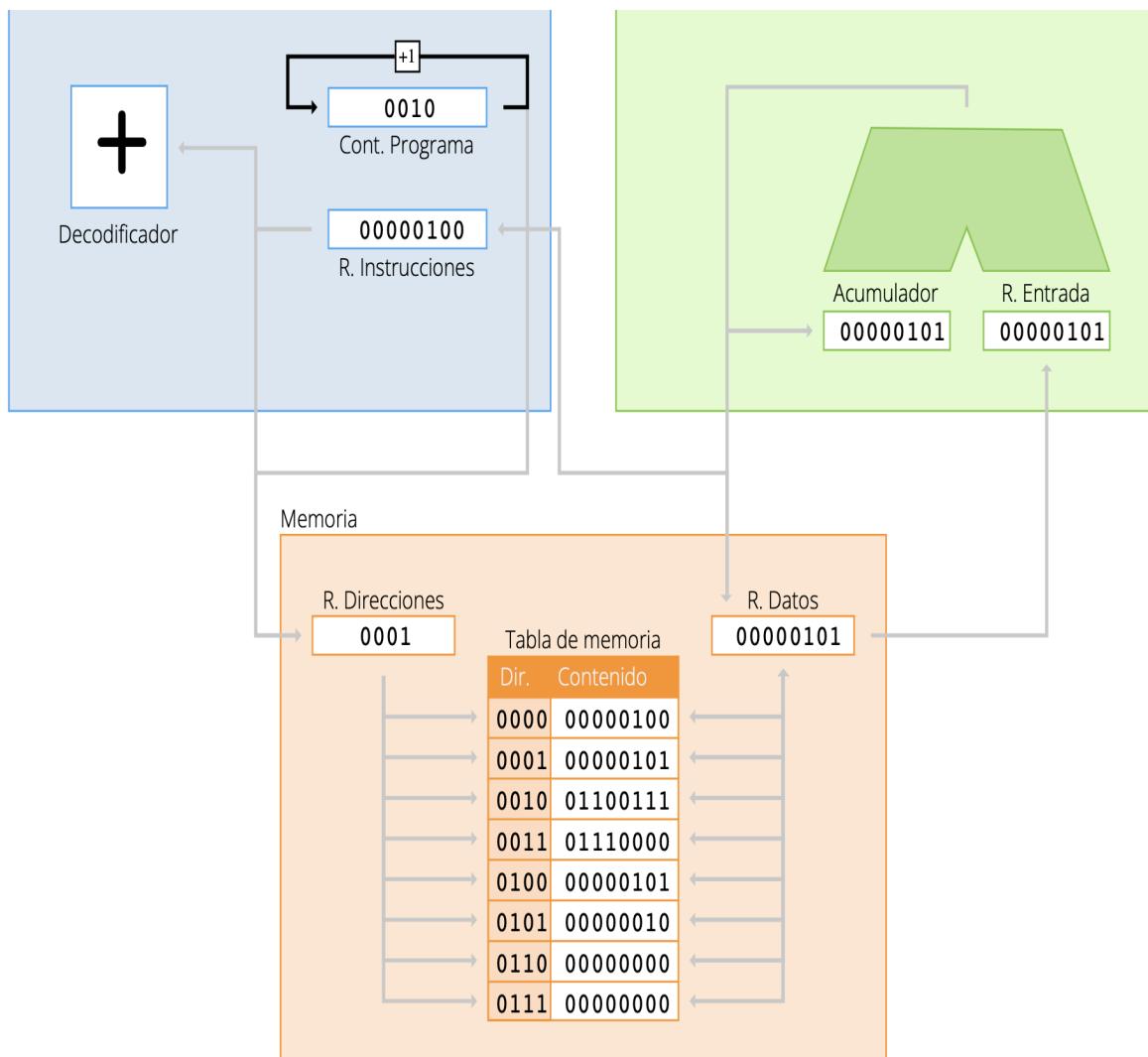
### Ejemplo de la acción número 12.

12. La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.



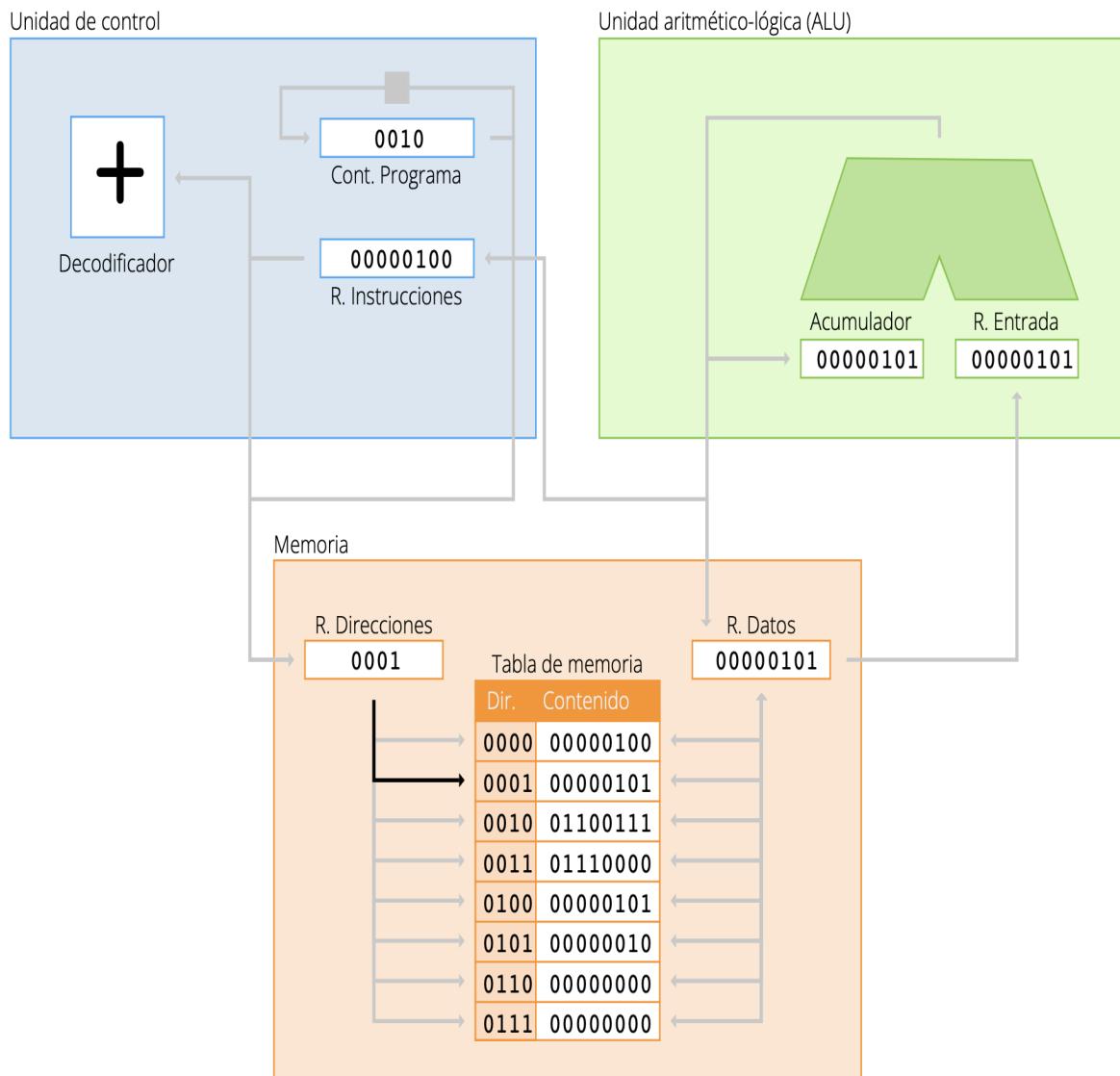
### Ejemplo de la acción número 13.

13. El contador de programa aumenta en 1, por lo que su contenido será la próxima instrucción a ejecutar.



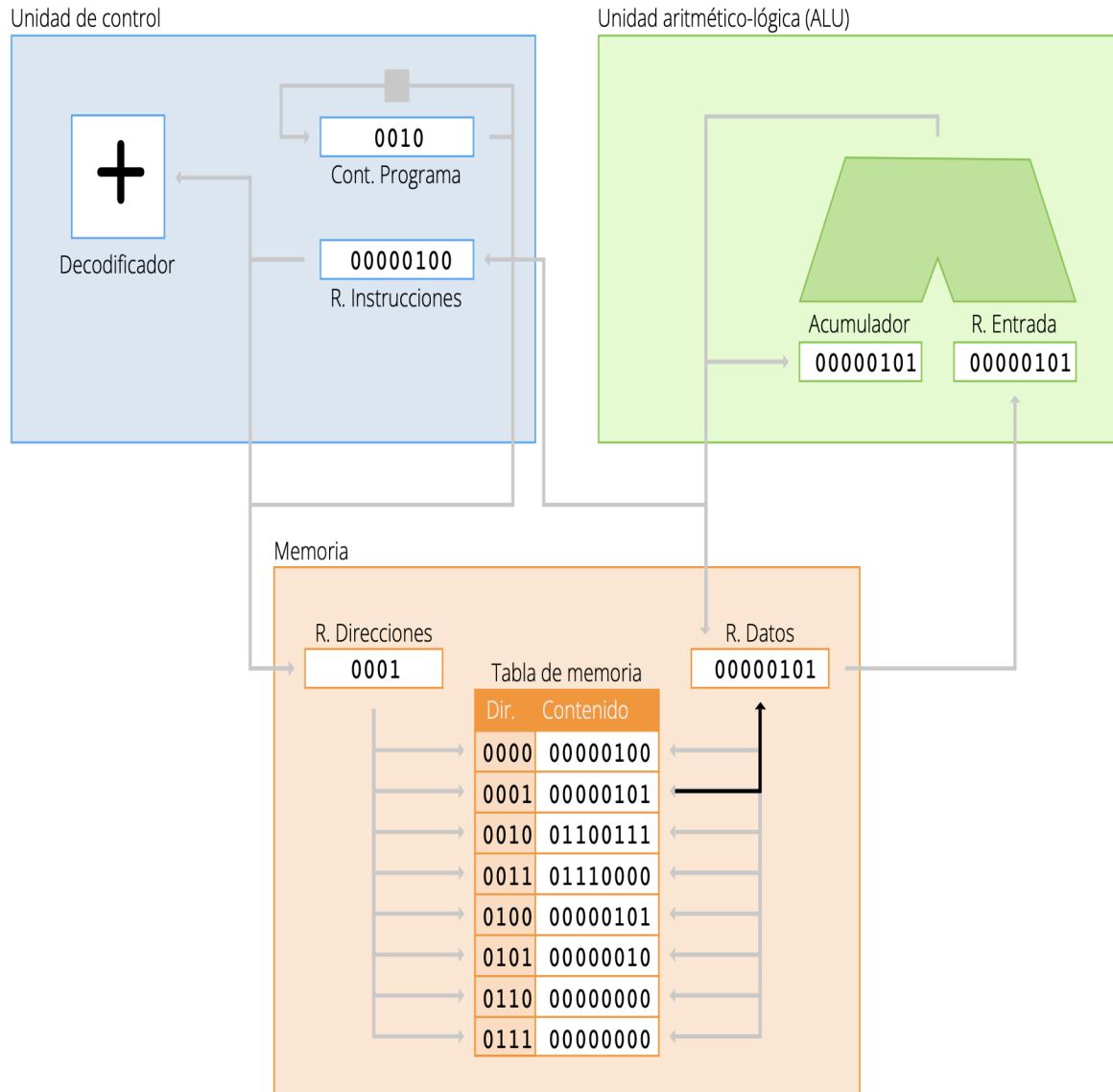
### Ejemplo de la acción número 14.

14. Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.



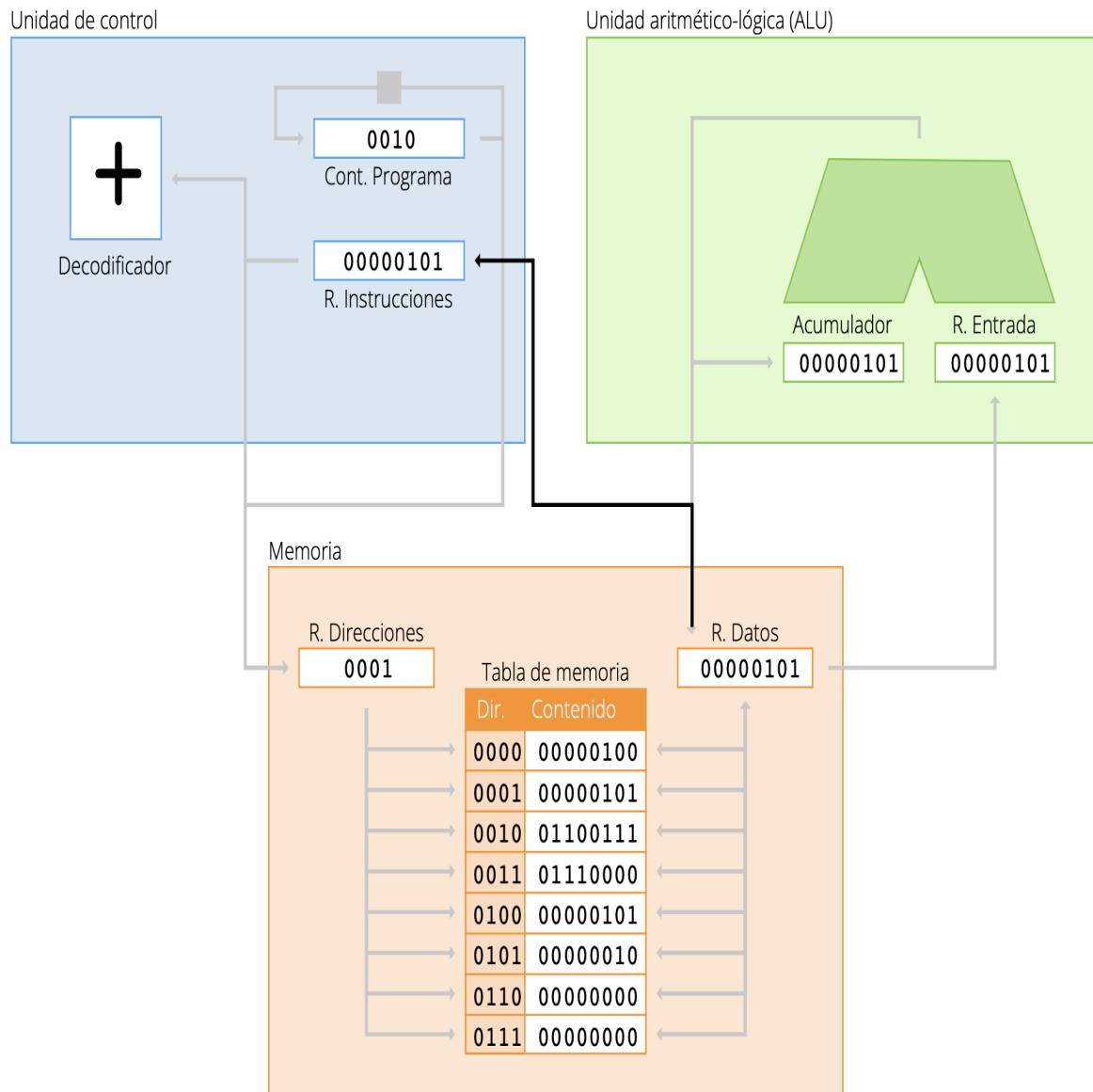
### Ejemplo de la acción número 15.

15. Se deposita en el registro de datos la acción a ejecutar.



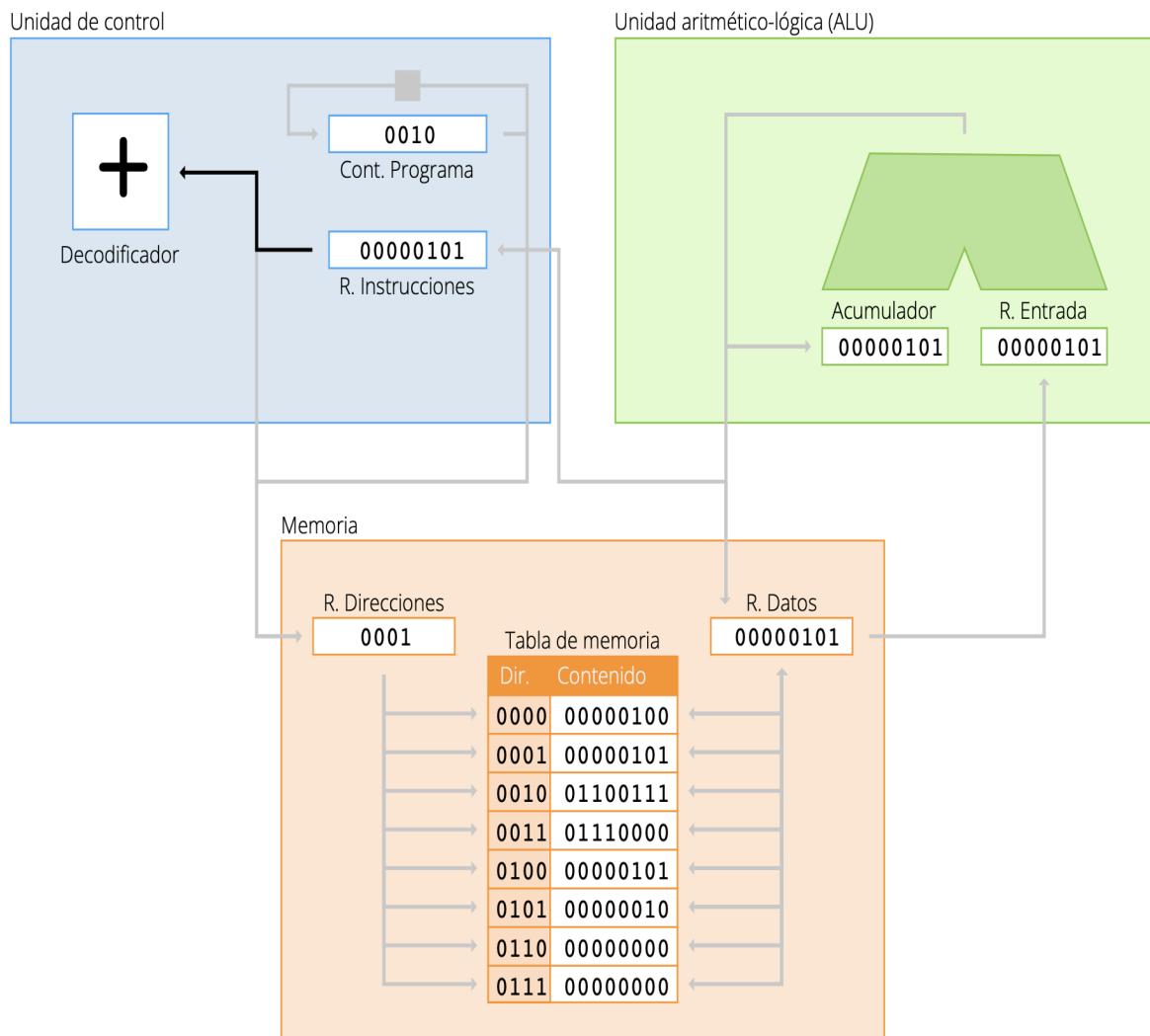
### Ejemplo de la acción número 16.

16. Se realiza el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.



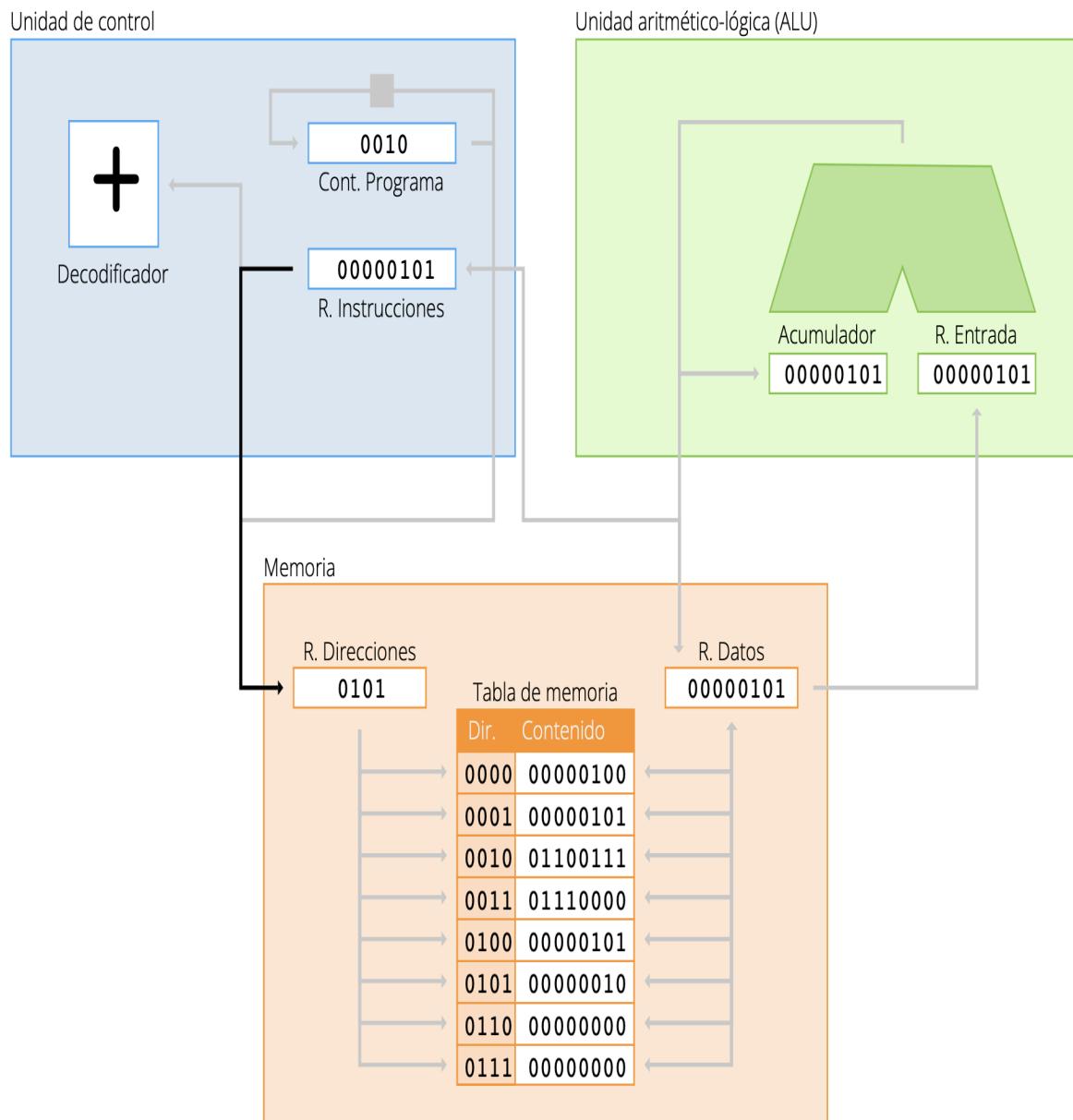
### Ejemplo de la acción número 17.

17. El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, lo cual quiere decir que interpreta el código de operación.



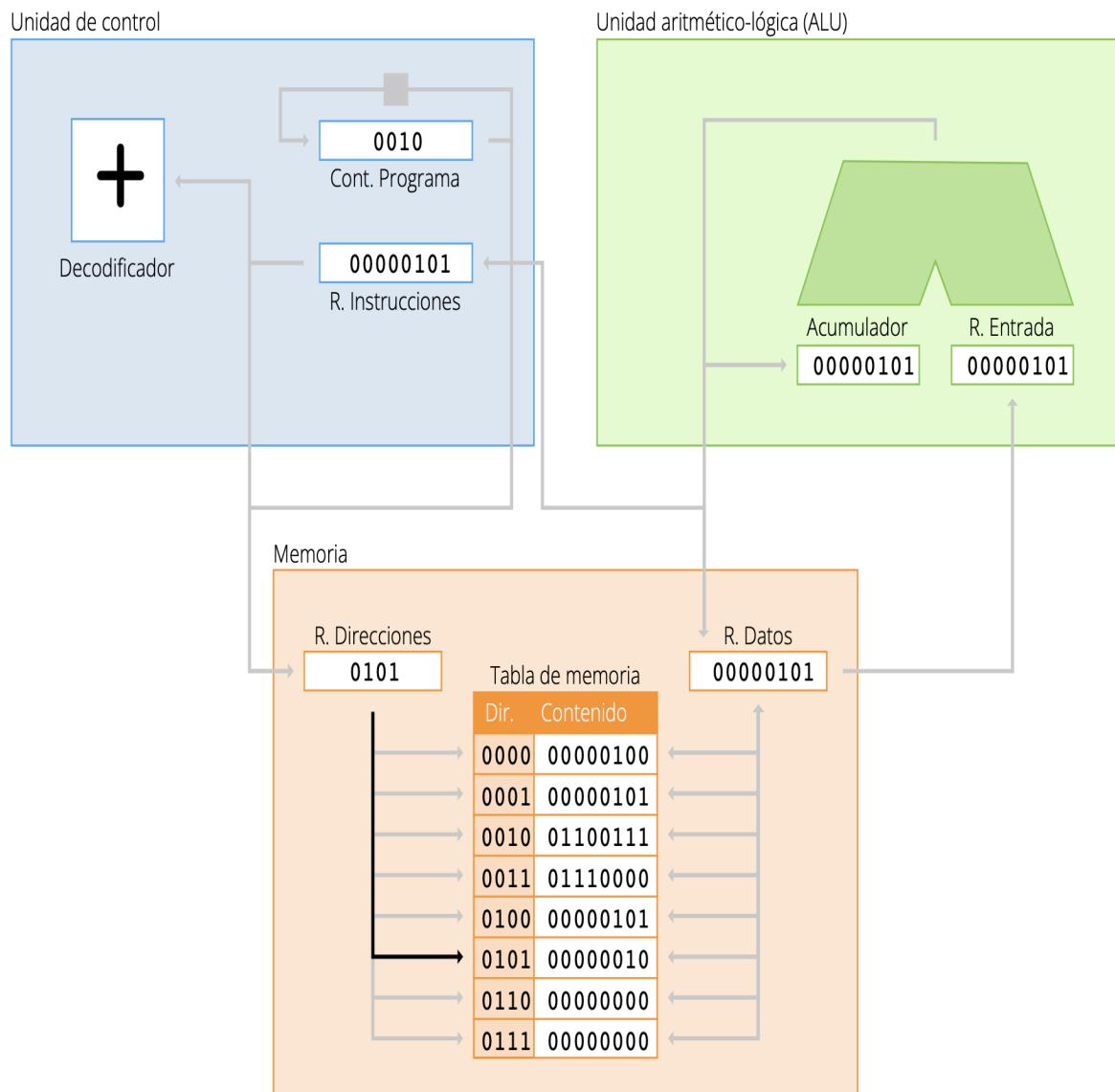
### Ejemplo de la acción número 18.

18. El registro de instrucciones envía los últimos cuatro bits al registro de direcciones.



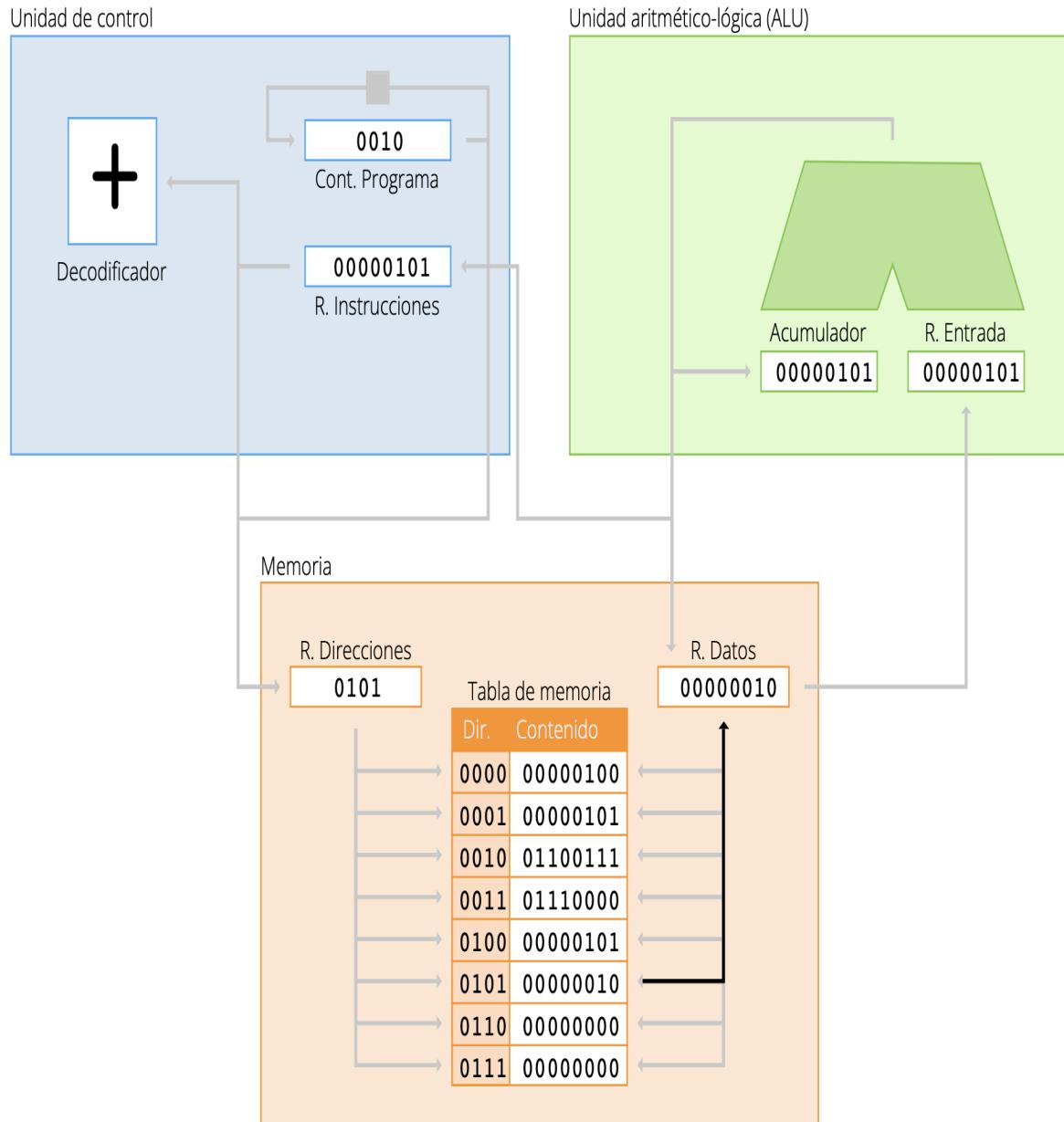
### Ejemplo de la acción número 19.

19. El registro de direcciones busca en la memoria la celda correspondiente y lee el dato.



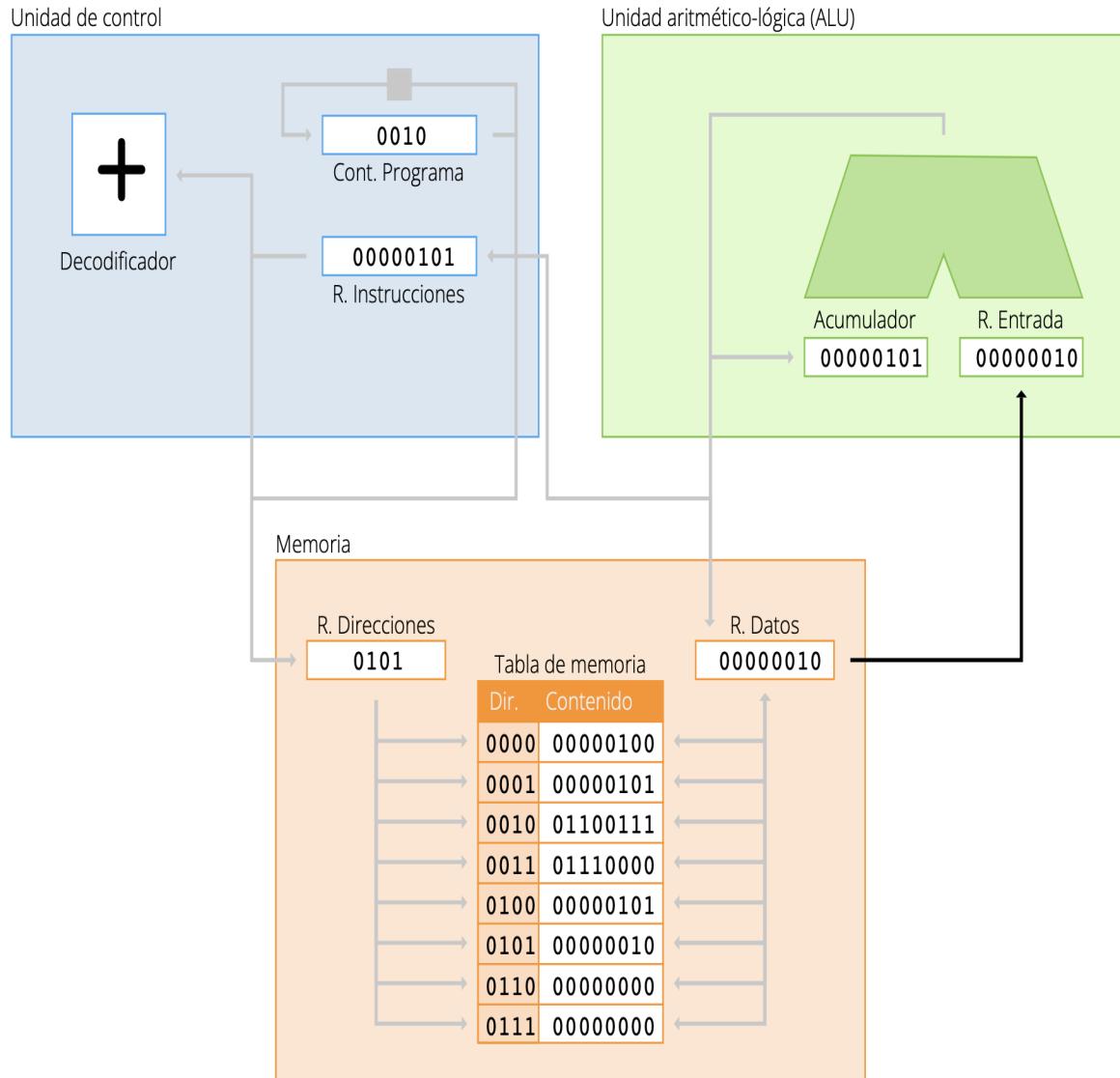
## Ejemplo de la acción número 20.

20. La información es enviada al registro de datos.



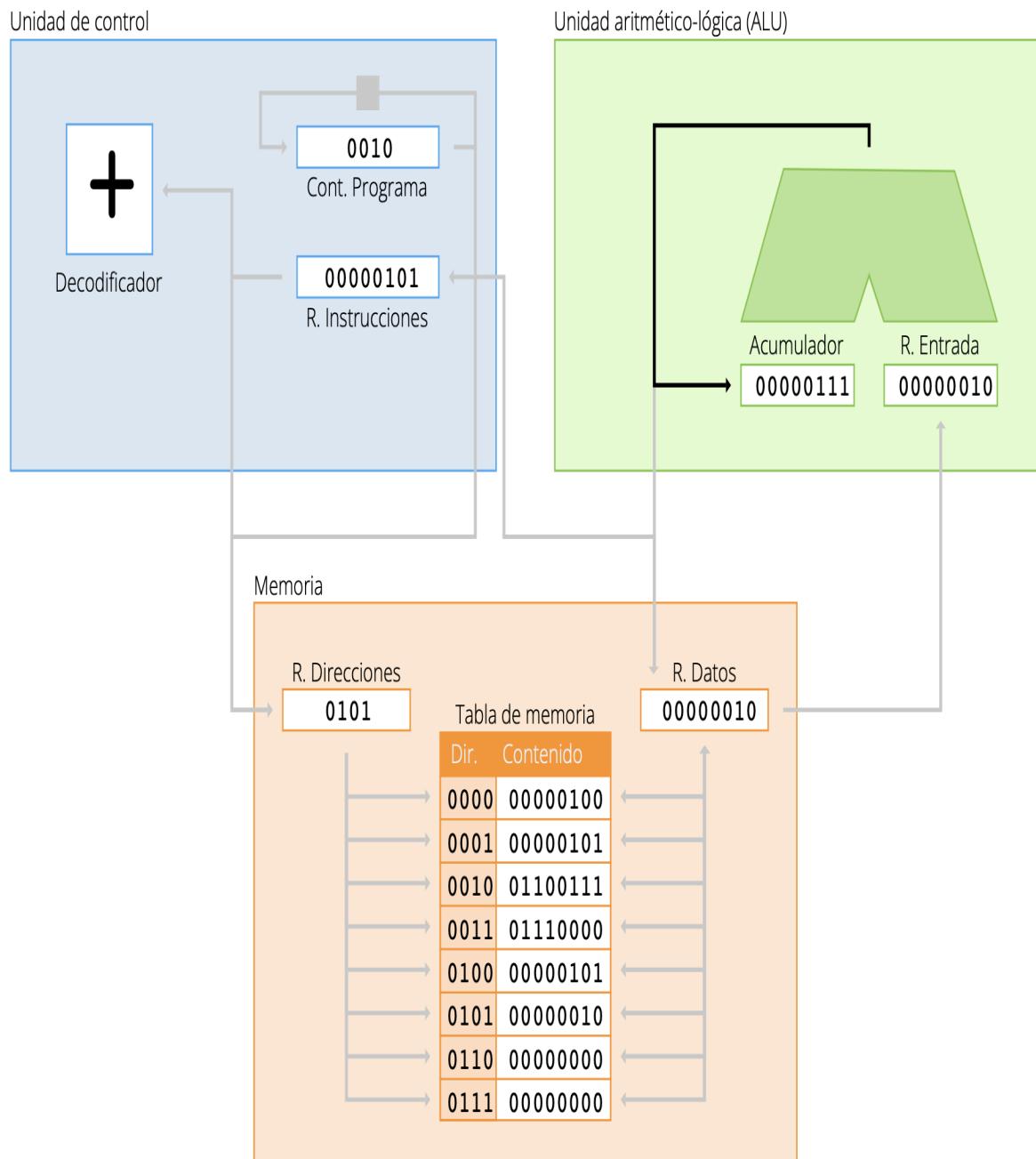
## Ejemplo de la acción número 21.

21. El registro de datos envía la información al registro de entrada.



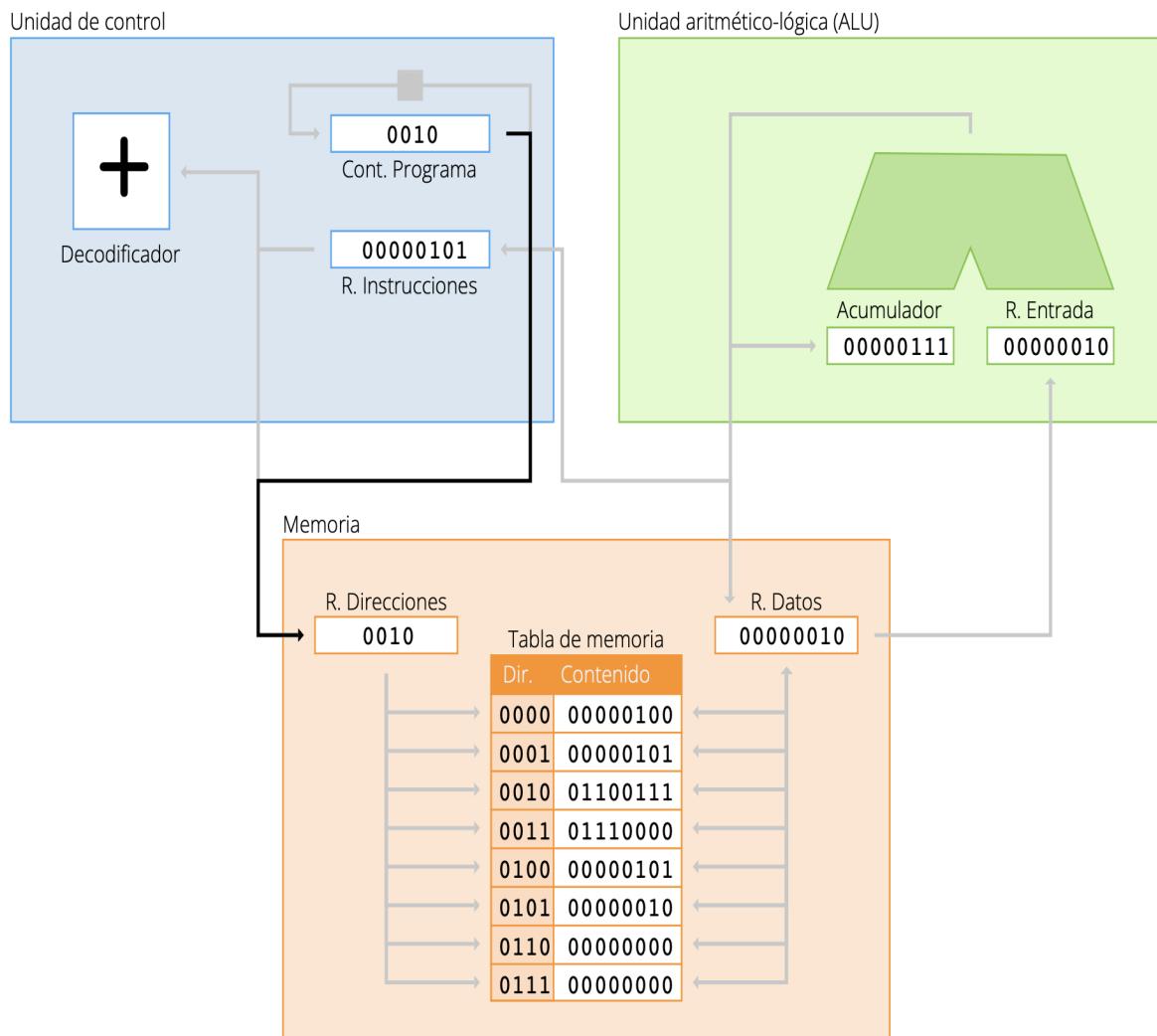
### Ejemplo de la acción número 22.

22. El circuito operacional realiza la acción con el registro acumulador y el registro de entrada, y lo almacena otra vez en el registro acumulador.



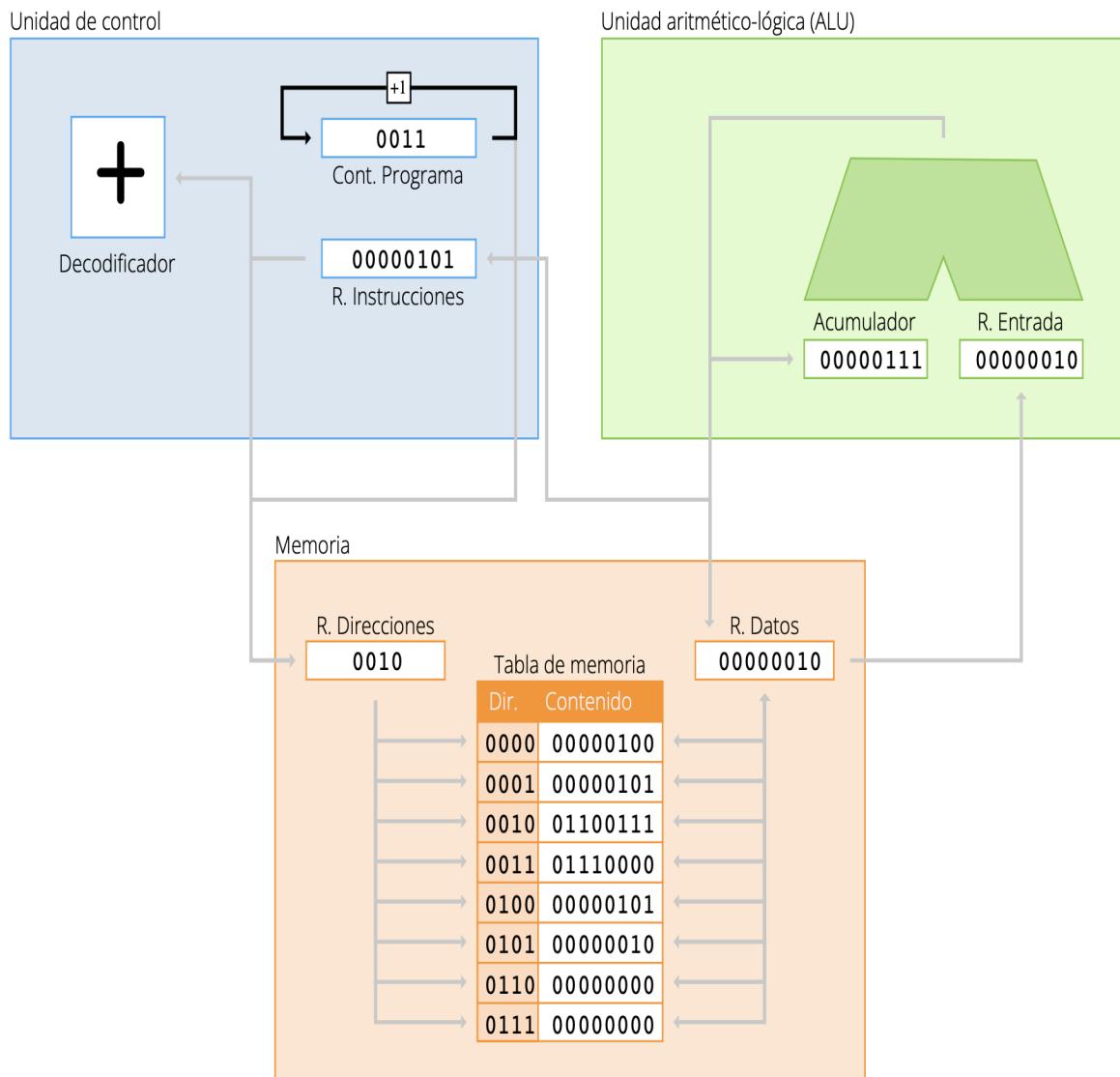
### Ejemplo de la acción número 23.

23. La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.



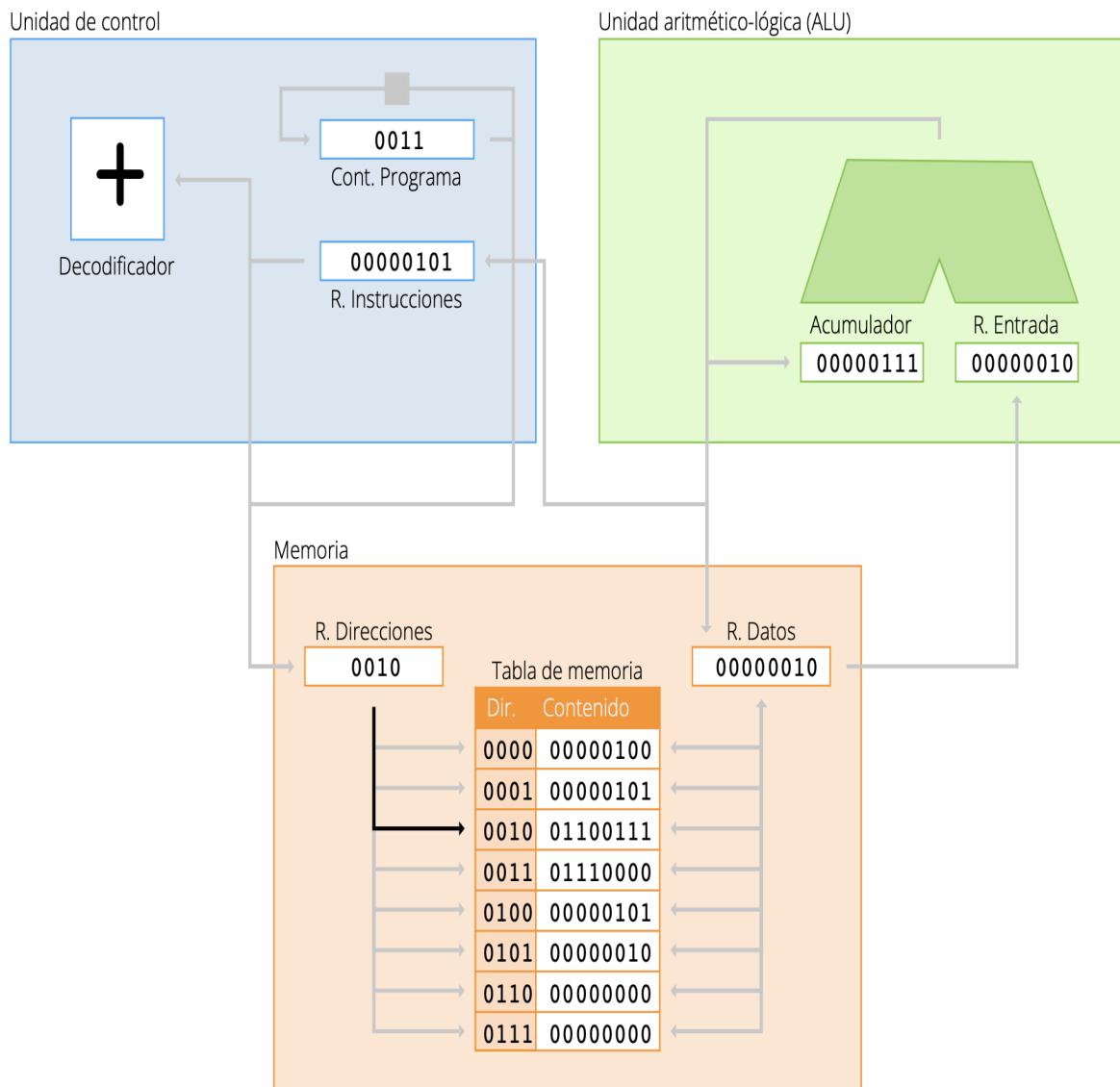
### Ejemplo de la acción número 24.

24. El contador de programas aumenta en uno, por lo que su contenido será la dirección de la próxima instrucción a ejecutar.



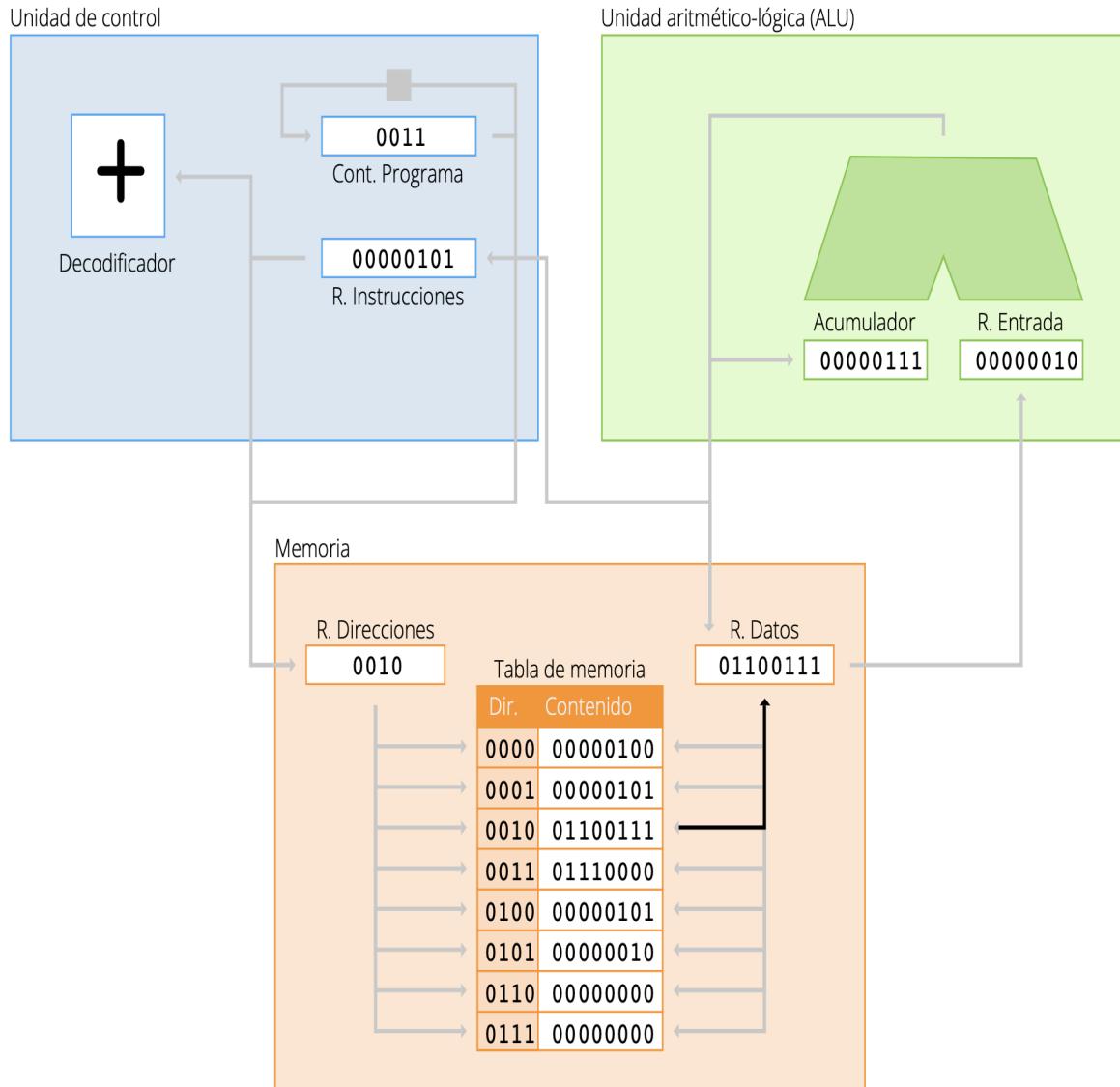
### Ejemplo de la acción número 25.

25. Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.



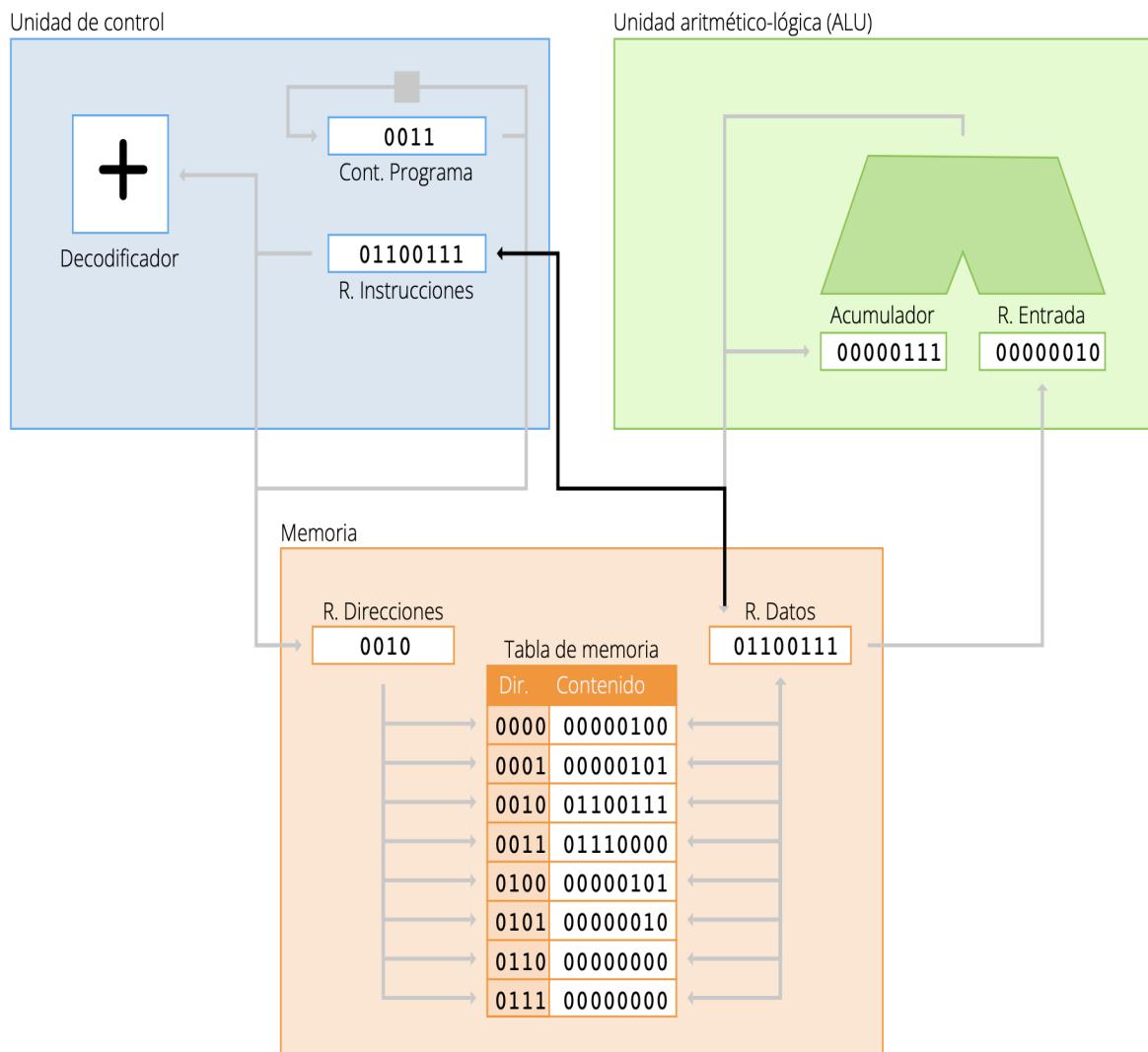
### Ejemplo de la acción número 26.

26. Se deposita en el registro de datos la instrucción a ejecutar.



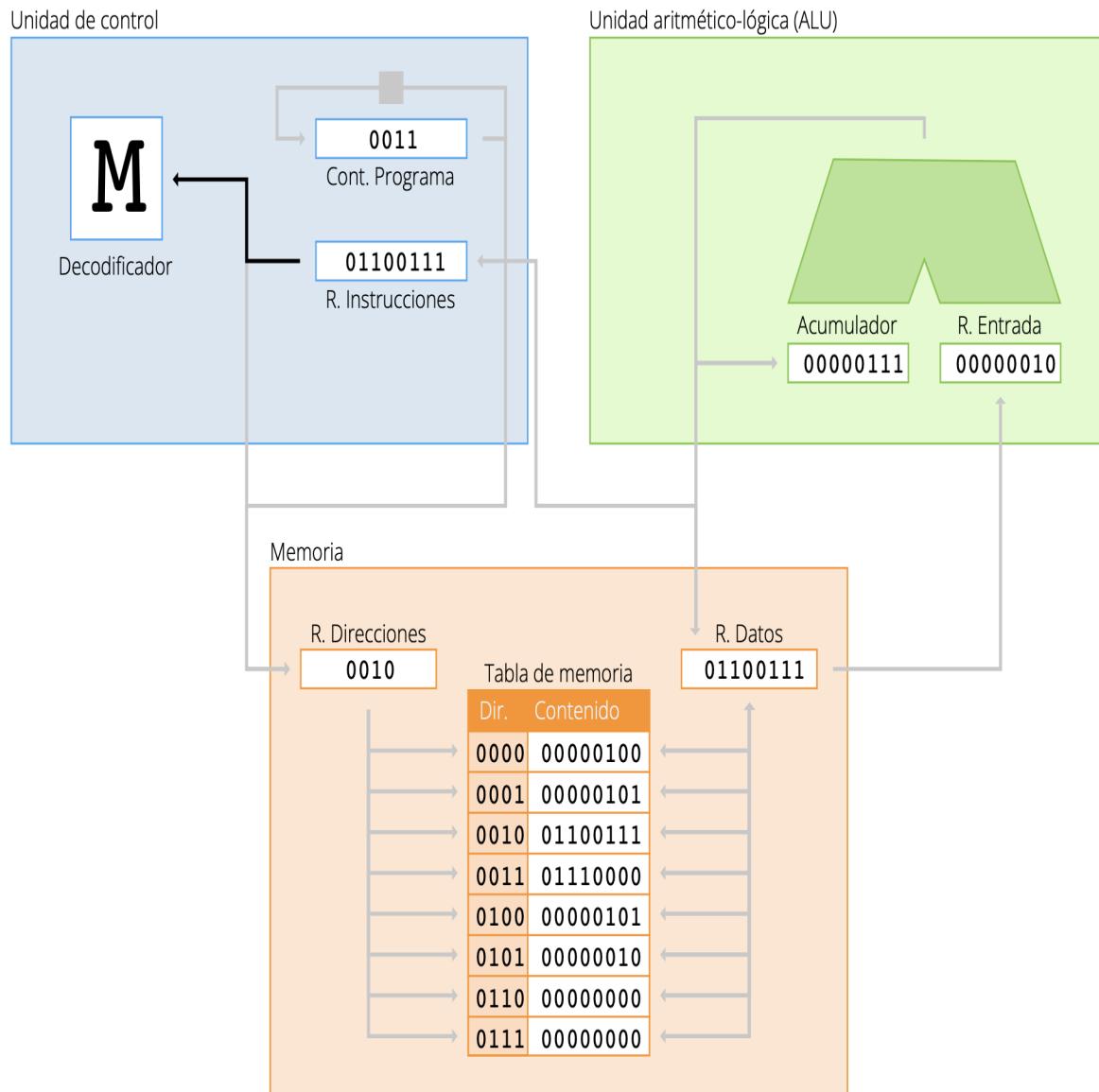
### Ejemplo de la acción número 27.

27. Se realizará el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.



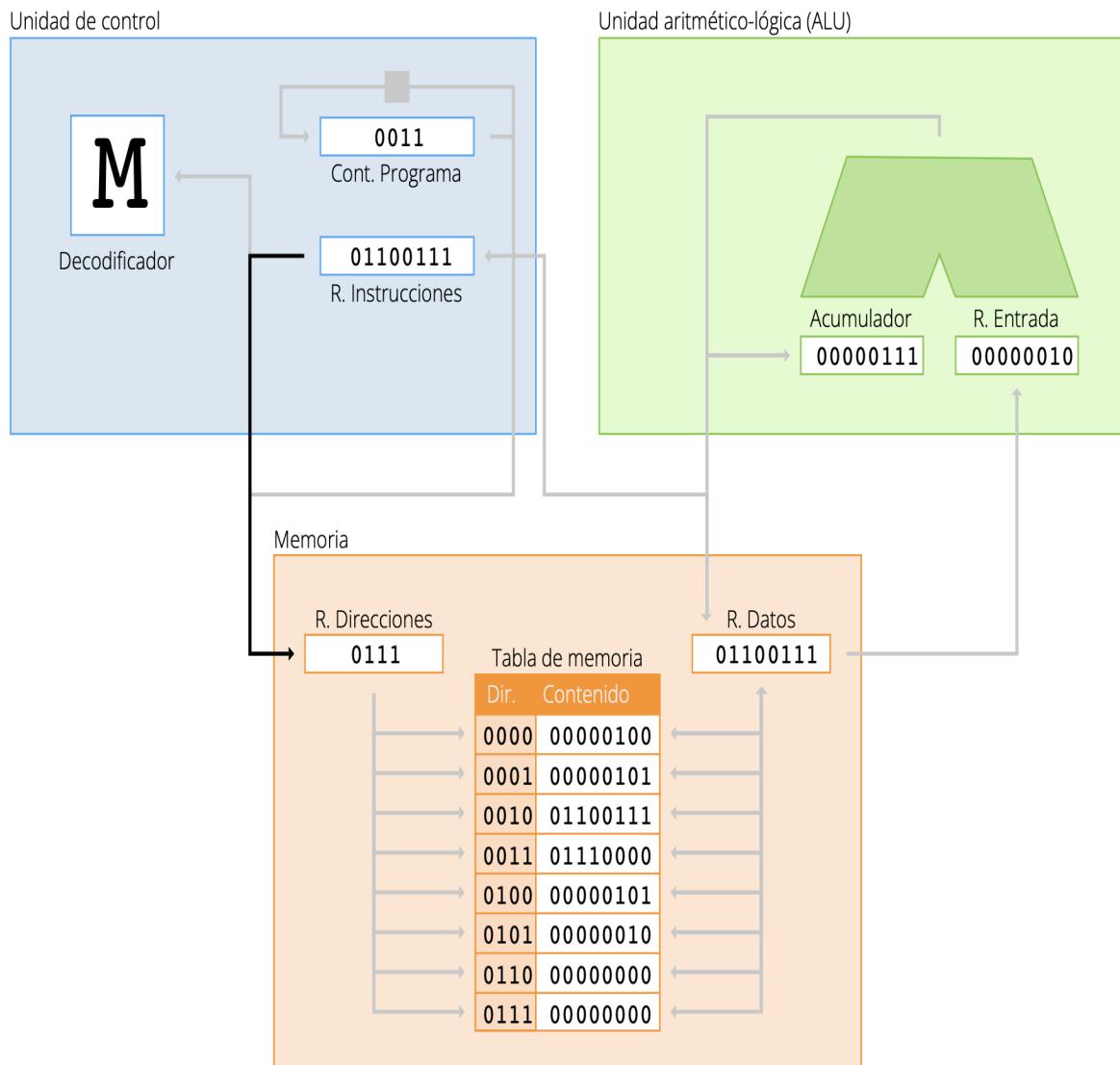
### Ejemplo de la acción número 28.

28. El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, es decir, que interpreta el código de la operación.



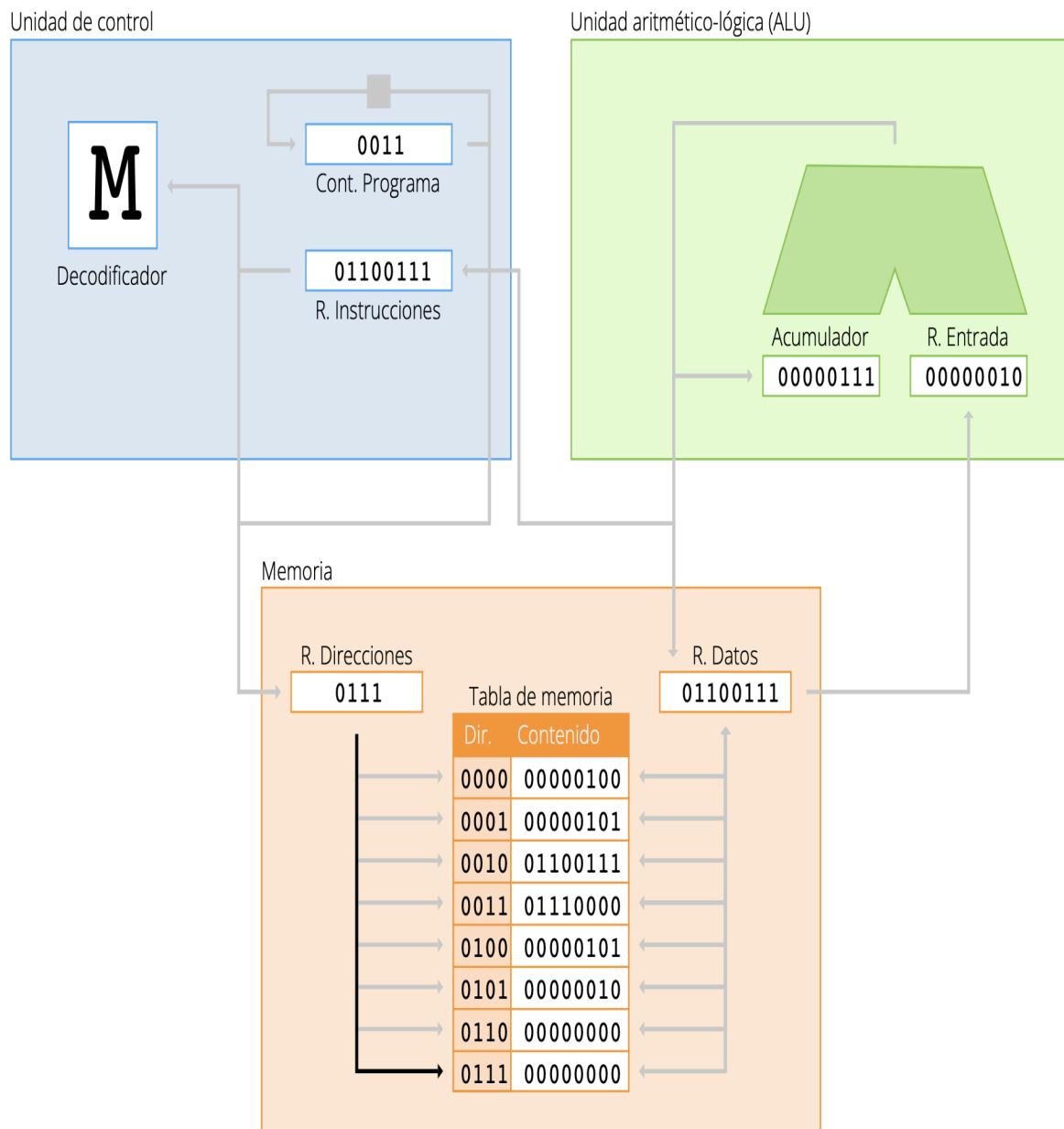
### Ejemplo de la acción número 29.

29. El registro de instrucciones envía los últimos cuatro bits al registro de direcciones.



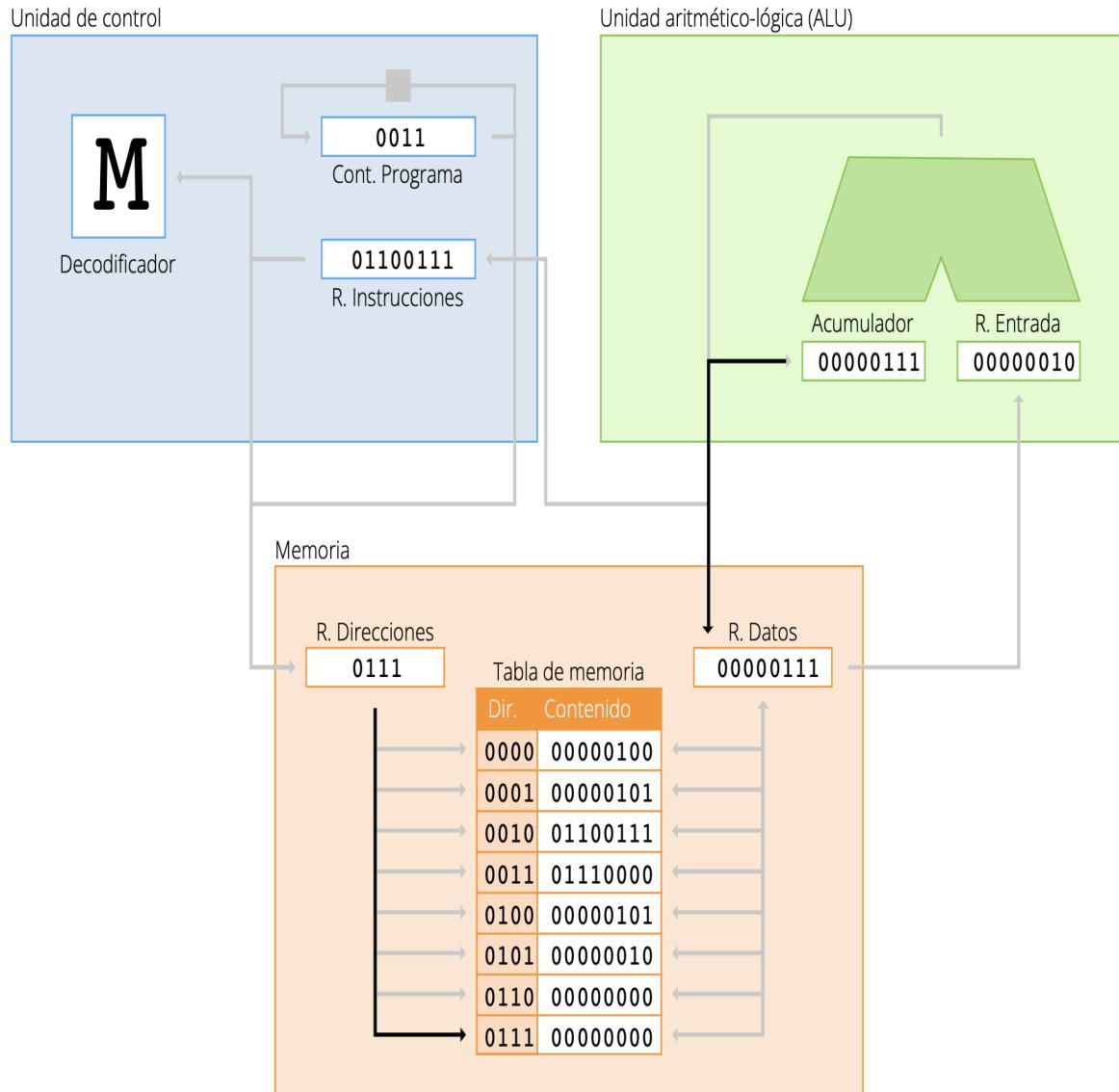
### Ejemplo de la acción número 30.

30. El registro de direcciones busca en la memoria en la celda en la que será guardada el resultado.



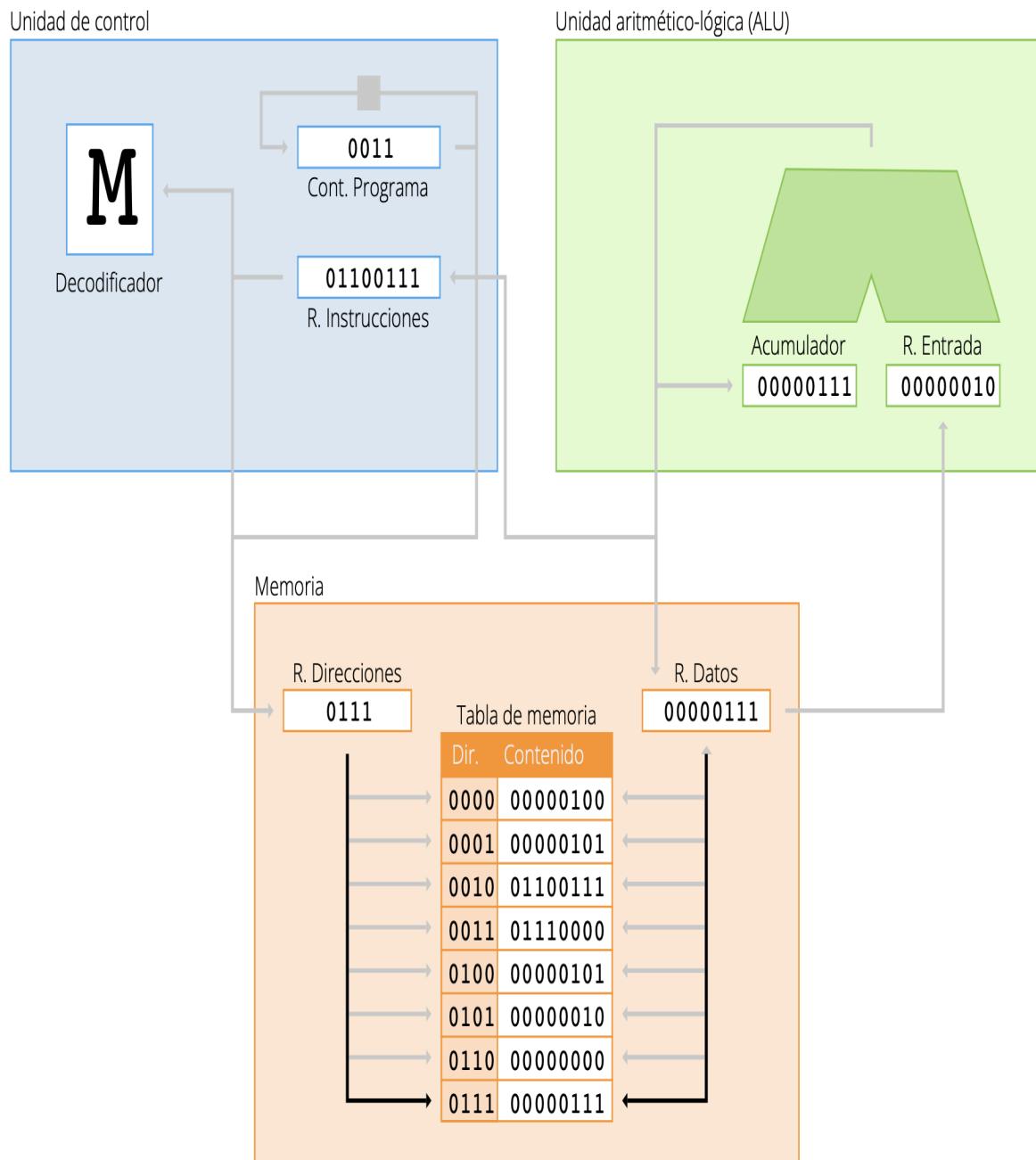
### Ejemplo de la acción número 31.

31. El registro acumulador envía la información al registro de datos.



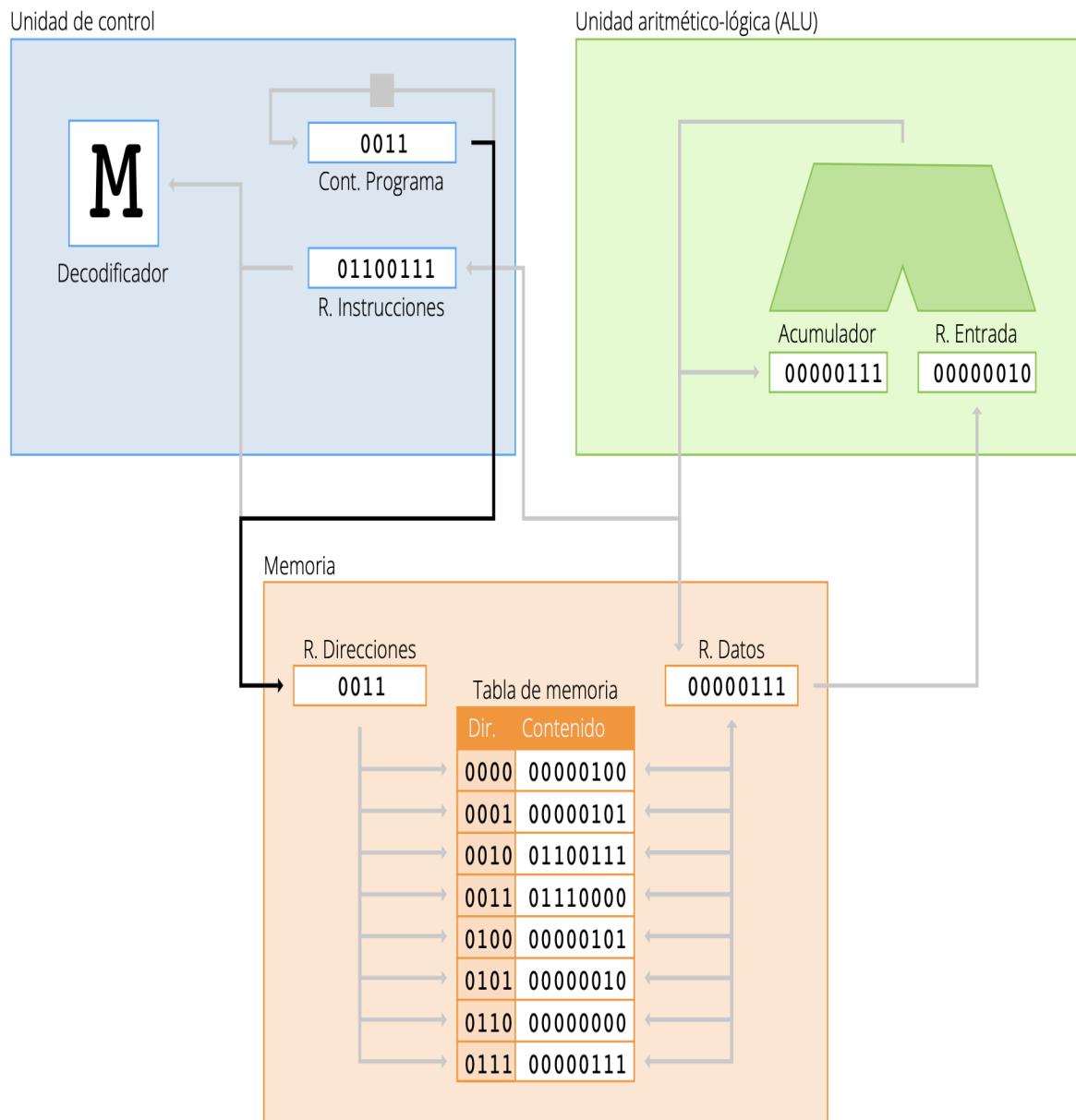
### Ejemplo de la acción número 32.

32. El registro de datos procede a la escritura de la información en la celda seleccionada por el registro de direcciones.



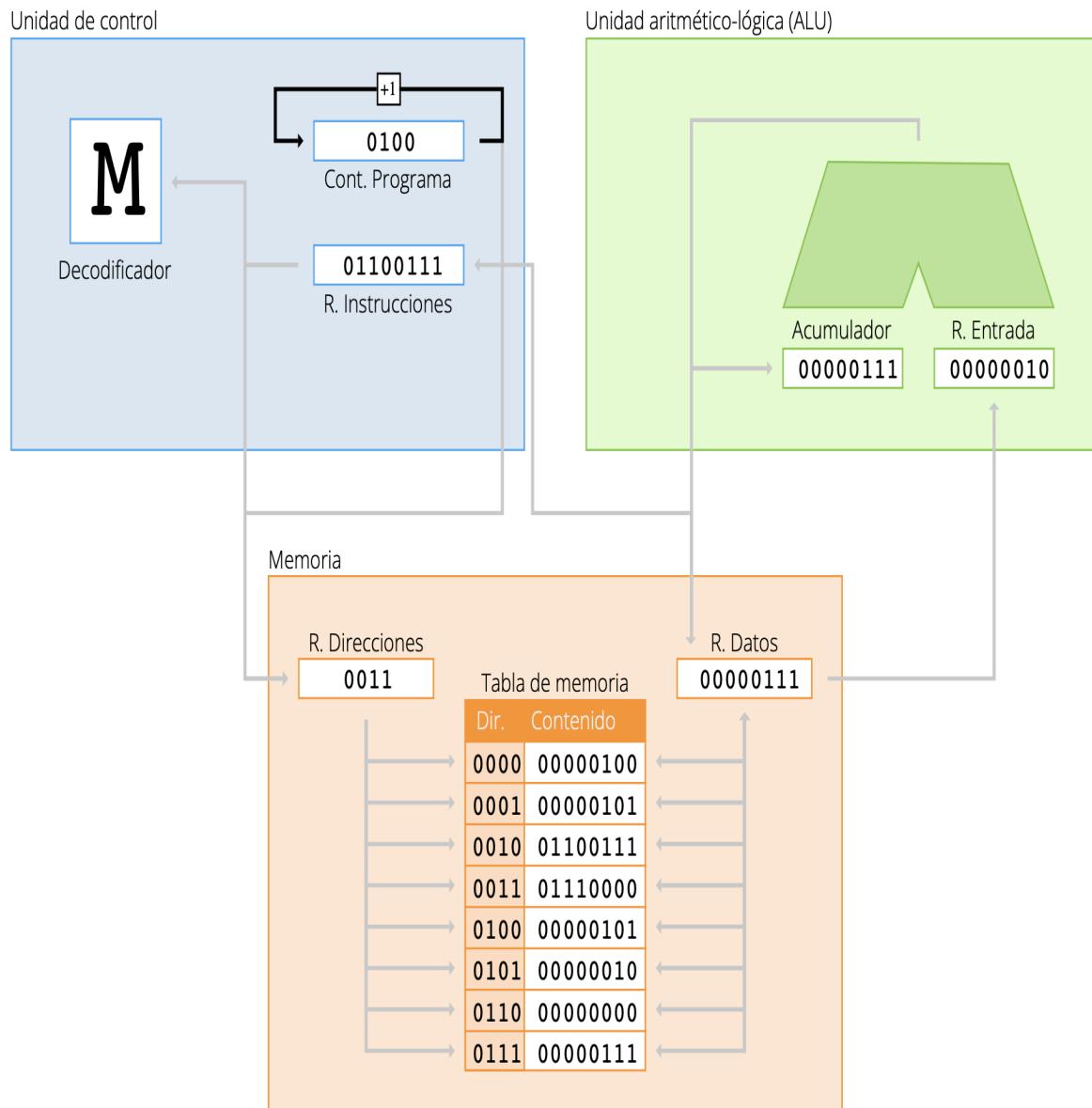
### Ejemplo de la acción número 33.

33. La unidad de control envía una micro-orden para transferir el contenido del contador de programa al registro de direcciones.



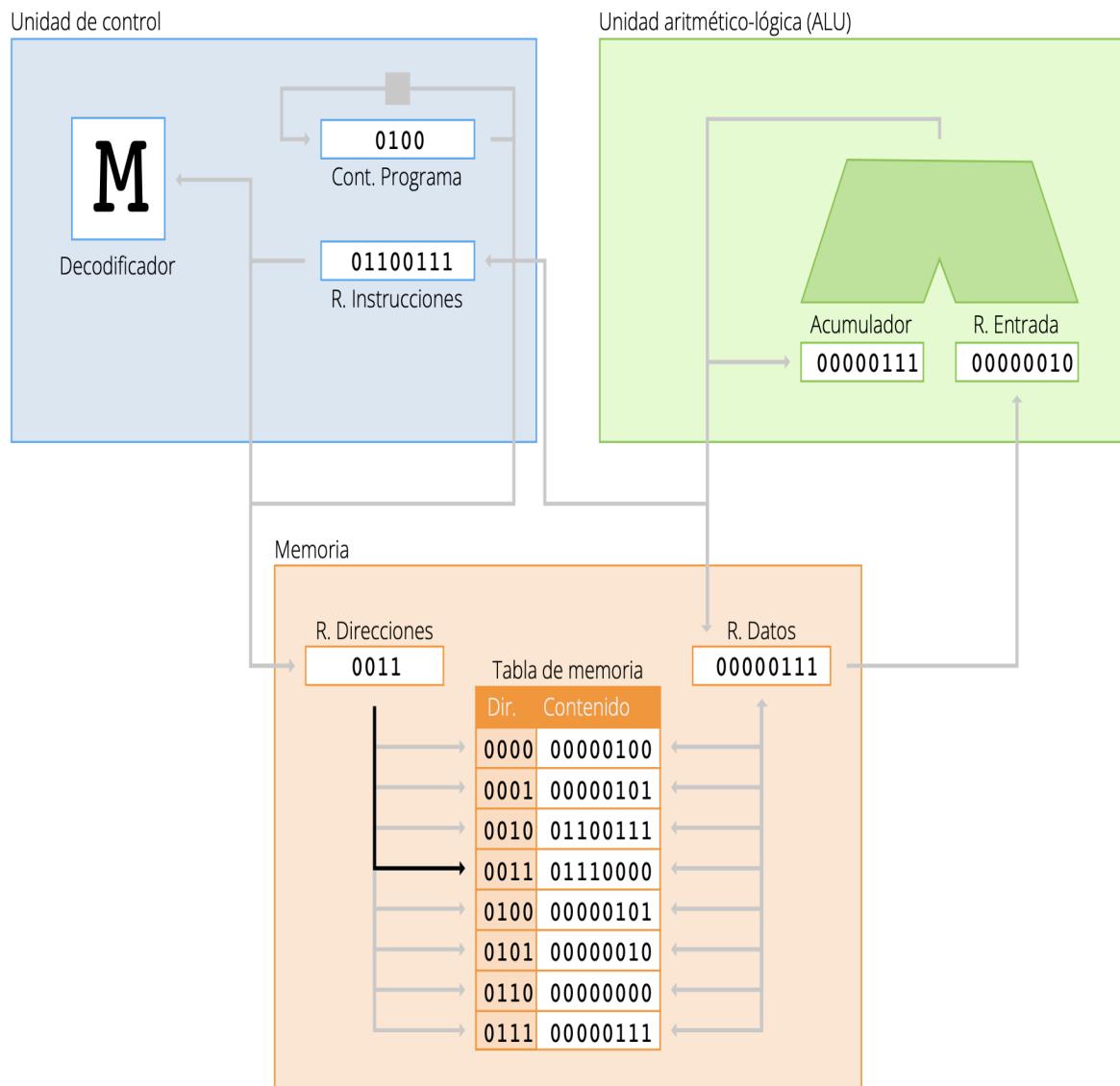
### Ejemplo de la acción número 34.

34. El contador de programa aumenta en uno, por lo que su contenido será la dirección de la próxima instrucción a ejecutar.



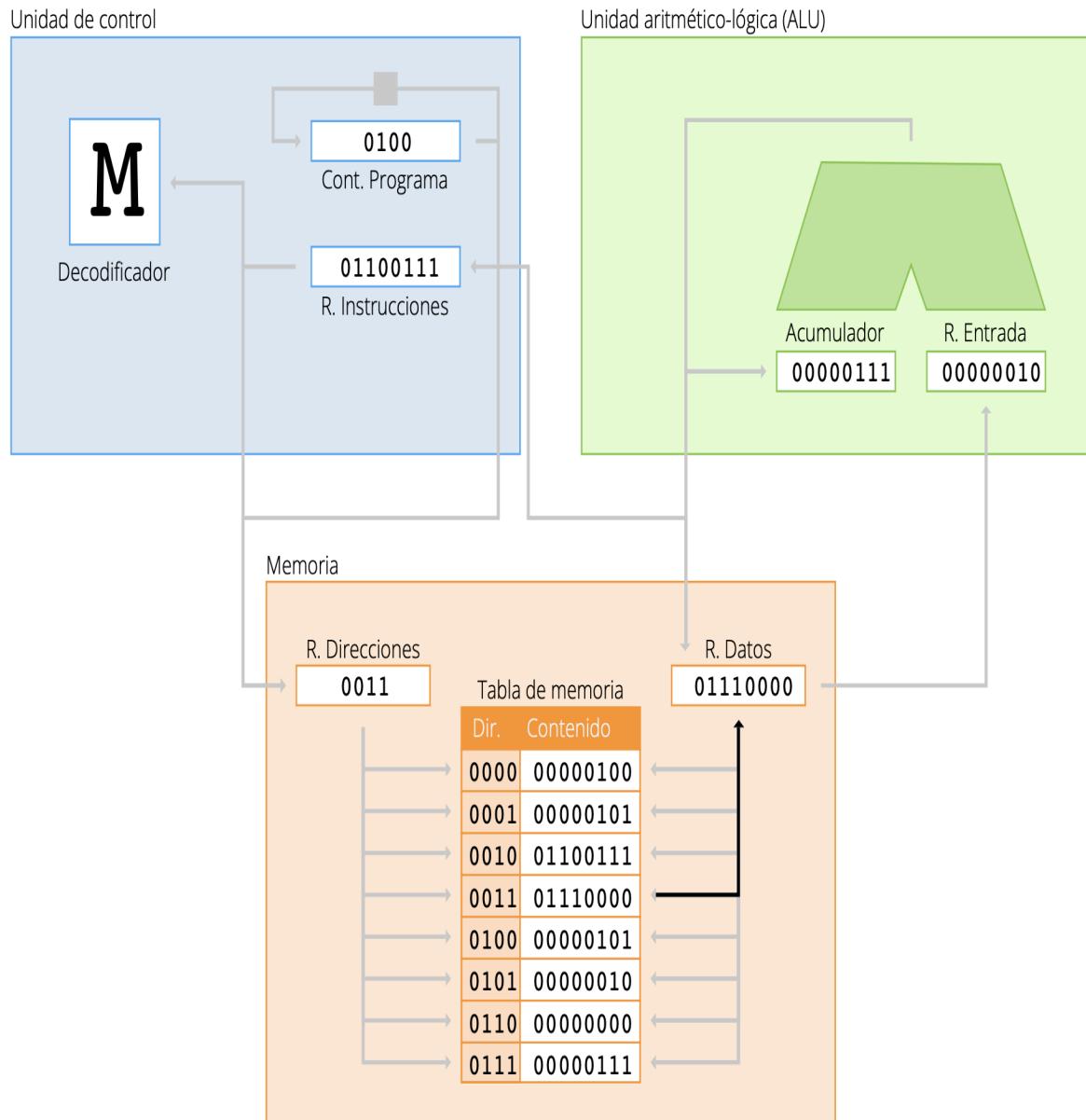
### Ejemplo de la acción número 35.

35. Se selecciona la posición de memoria que indica el registro de direcciones y se realiza una lectura en la memoria.



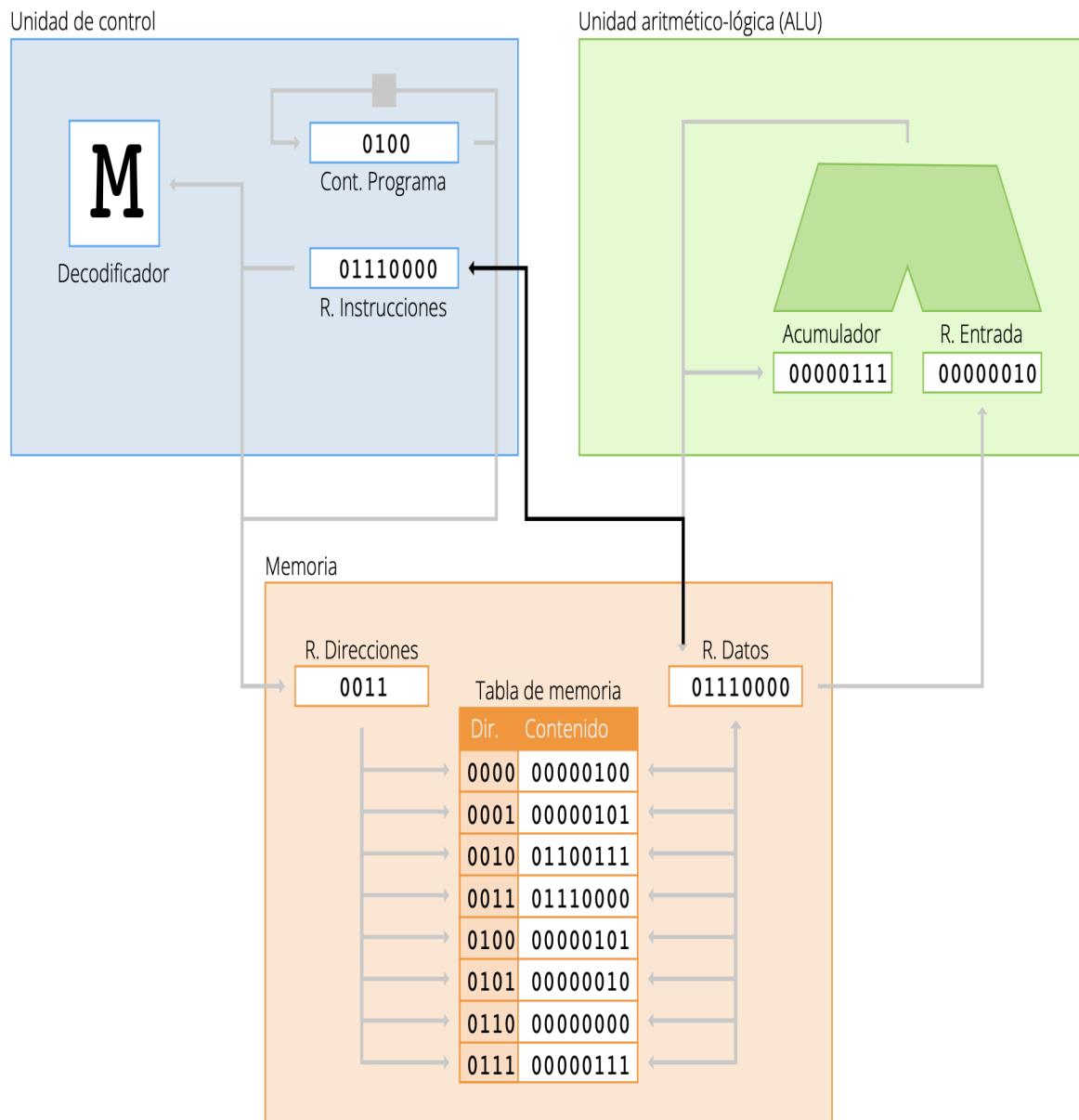
### Ejemplo de la acción número 36.

36. Se deposita en el registro de datos la instrucción a ejecutar.



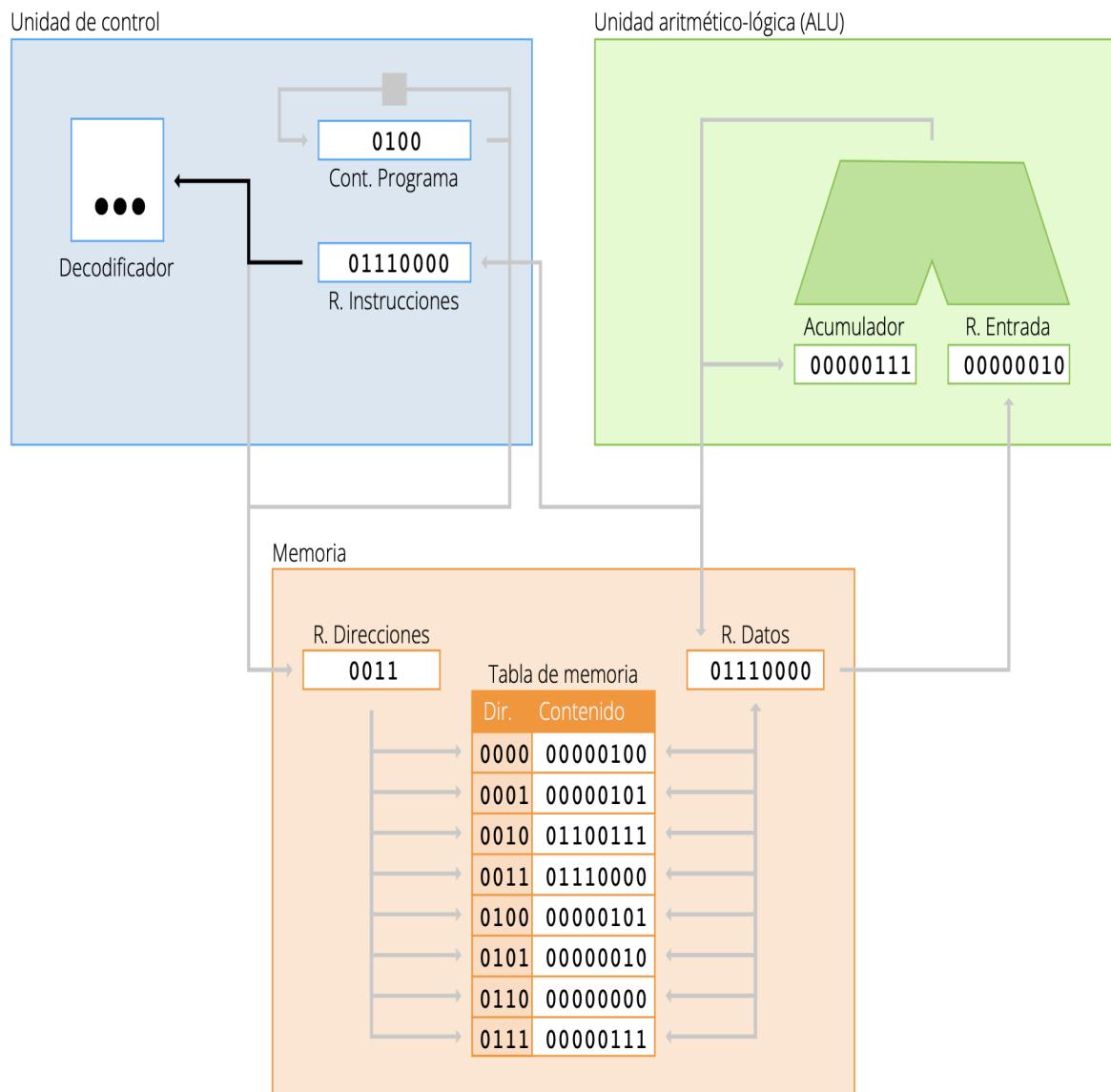
### Ejemplo de la acción número 37.

37. Se realiza el traslado de la información contenida en el registro de datos al registro de instrucciones, donde se almacenará.



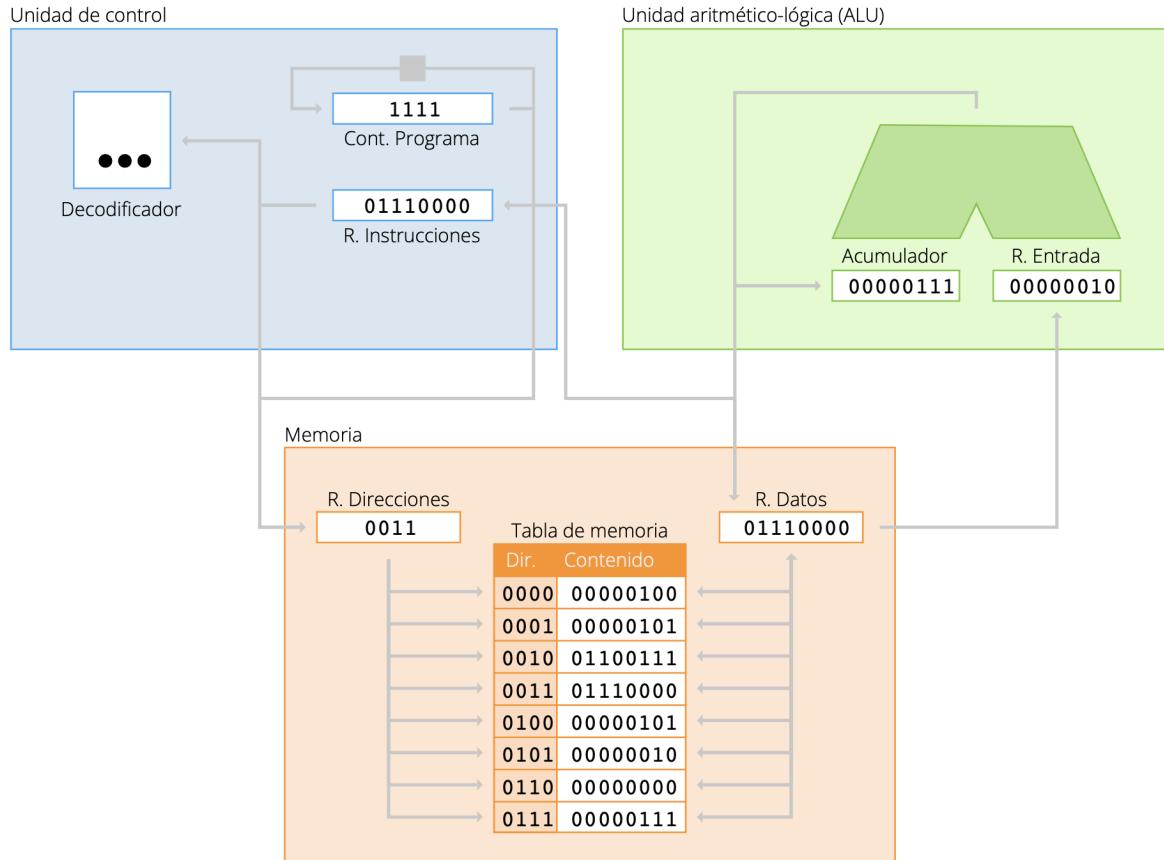
### Ejemplo de la acción número 38.

38. El decodificador procede a la interpretación de la instrucción que serán los cuatro primeros bits, es decir, que interpreta el código de operación.



## Ejemplo de la acción 39.

39. El decodificador interpreta que se finaliza el programa y se para la ejecución.



Si nos fijamos, en el acumulador, podemos ver el número binario 0111, que en decimal es igual a 7, lo que nos da el resultado de nuestra operación.