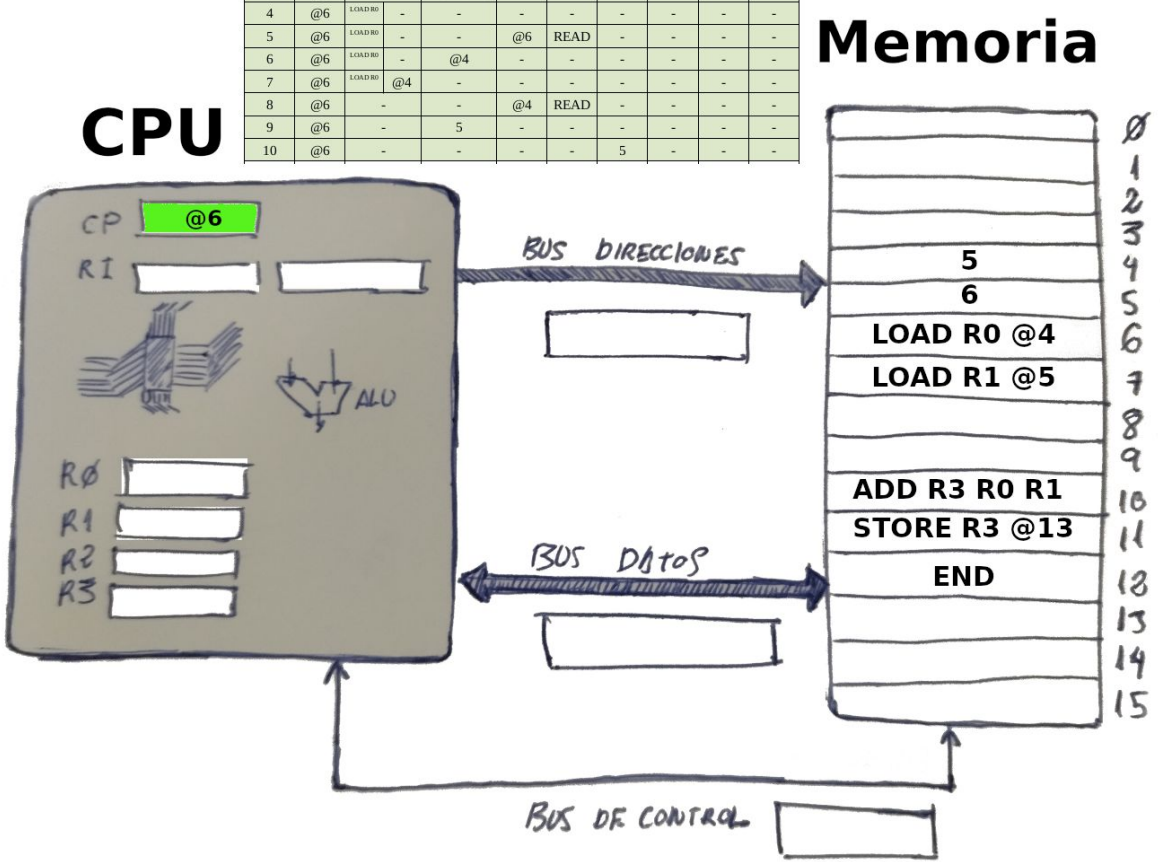
Ejecución de instrucciones

- CPU de 4 bits
- Direcciones de memoria de 8 bits
- Buses de 4 bits

Ciclo 1

- El **contador de programa** (**PC**) indica qué instrucción será la siguiente en ejecutarse
- La siguiente instrucción en ejecutarse será la que se almacena en la dirección 6 de la memoria (`@6`)

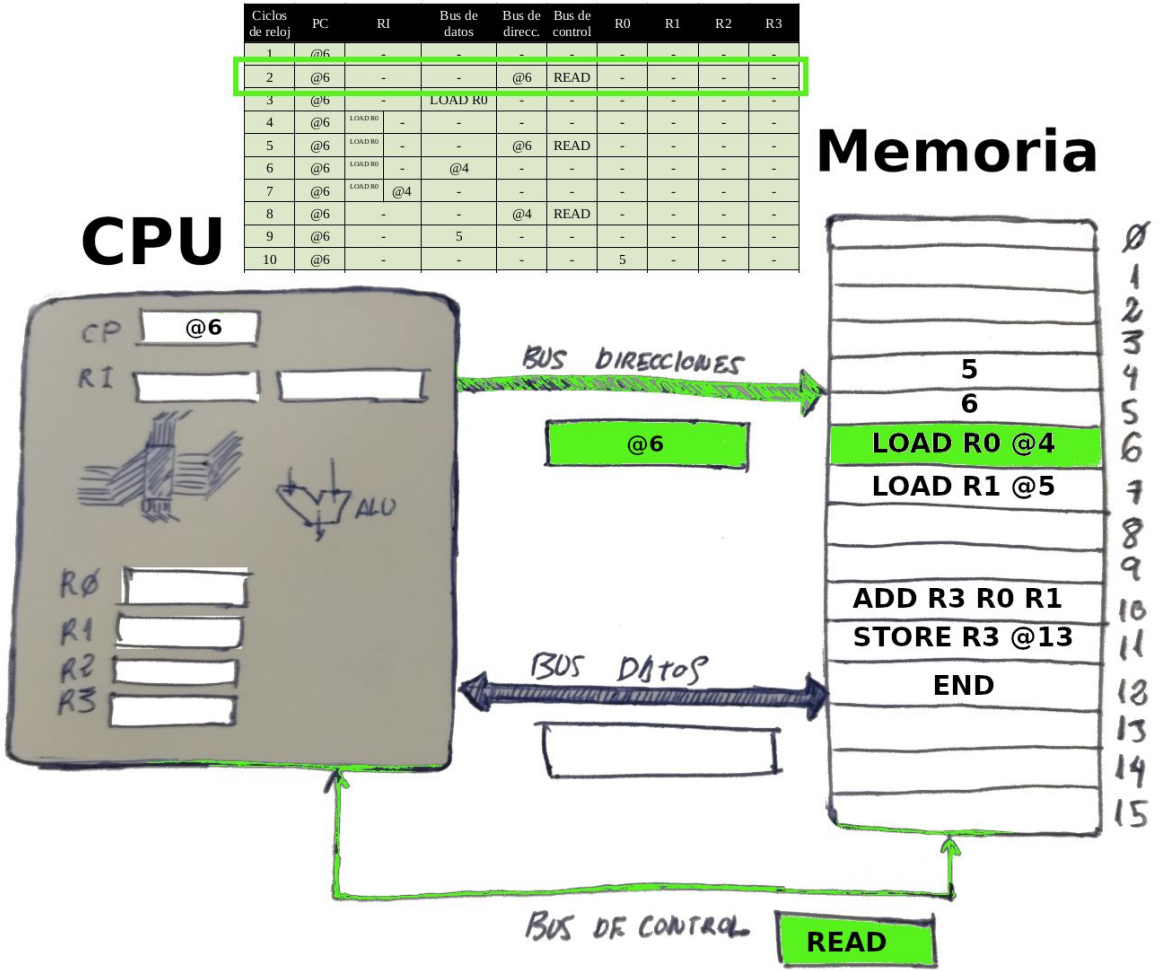
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 2

- La CPU indica al **bus de direcciones** que apunte a `@6`
- La CPU indica al **bus de control** que quiere leer el contenido de la dirección de memoria apuntada (`READ`)
- La **instrucción** que se leerá de memoria será: `LOAD R0 @4`

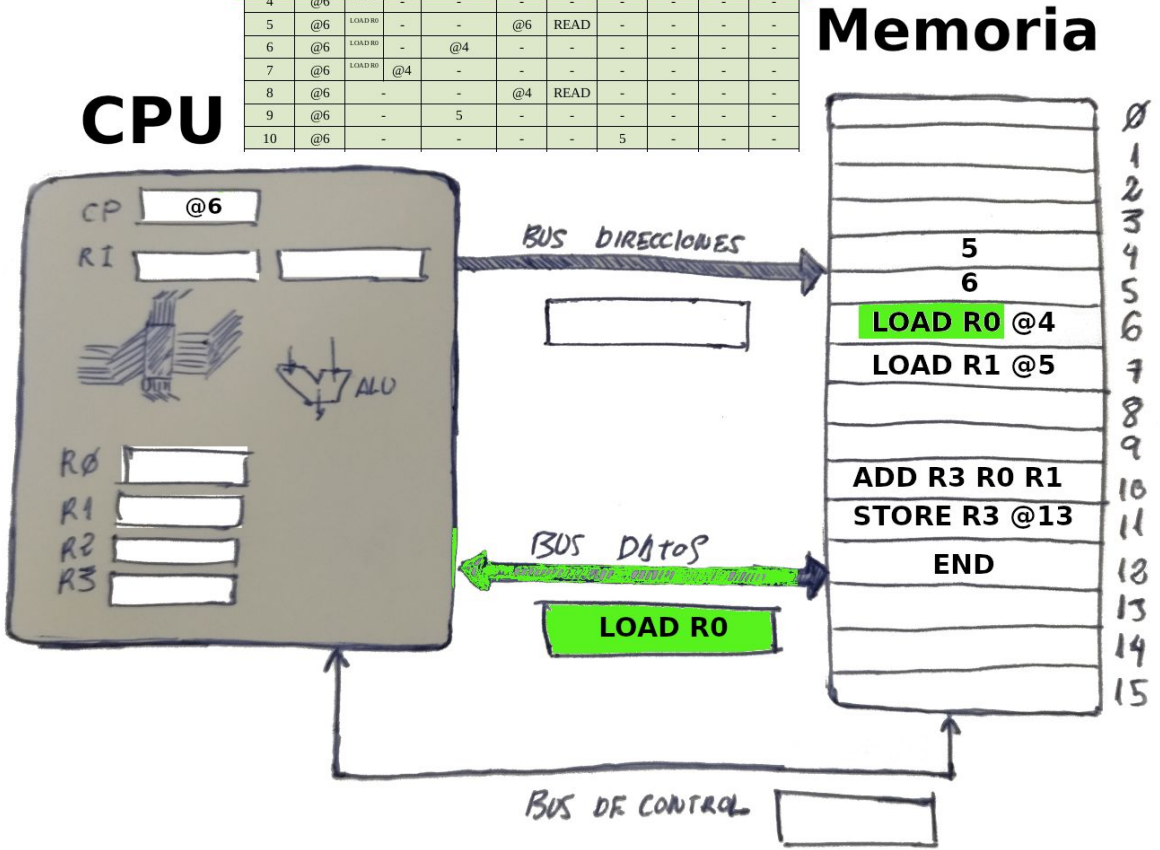
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 3

- Las instrucciones en la memoria ocupan 8 bits
- La CPU y los buses son solo de 4 bits
- La instrucción se tiene que obtener en 2 partes
- En esta **primera parte** se obtiene **`LOAD R0`**
- La memoria pasa **`LOAD R0`** al **bus de datos**

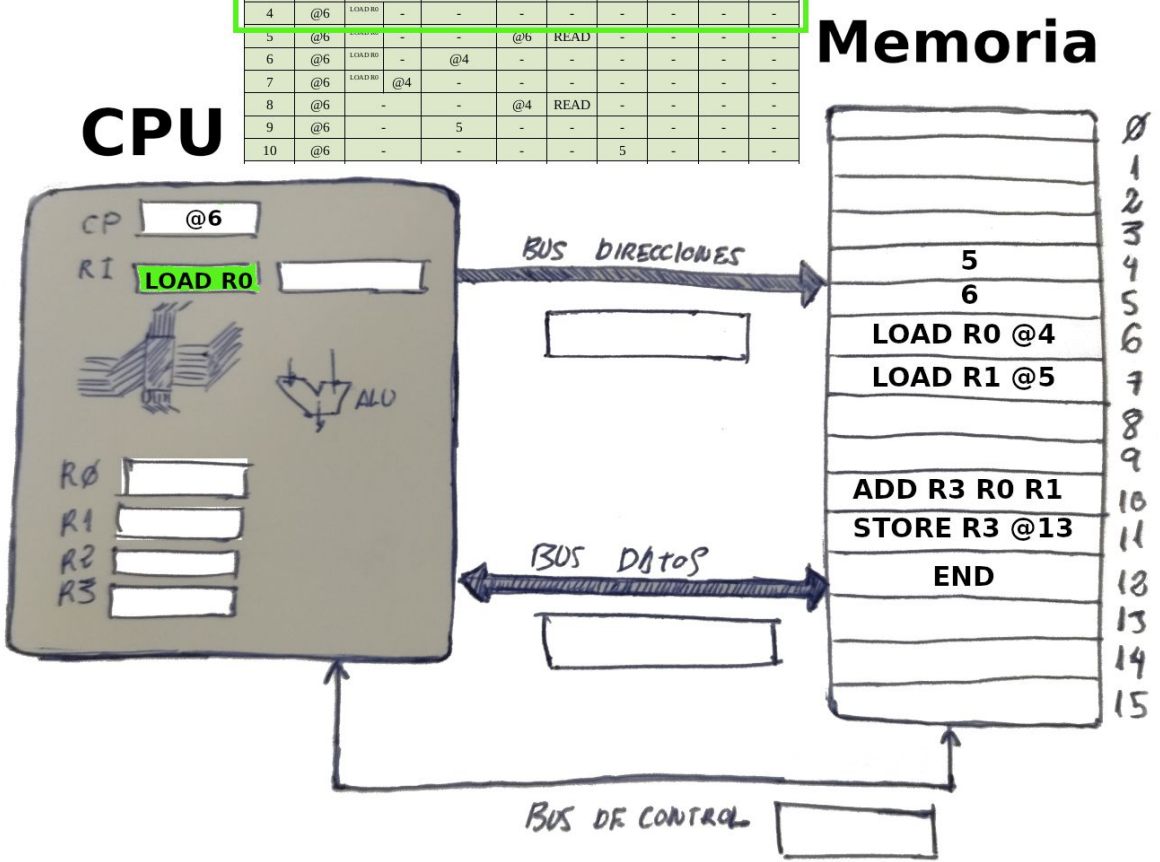
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	-	-	-	-	-	-	-	-
5	@6	LOAD R0	-	-	@6	READ	-	-	-
6	@6	LOAD R0	-	@4	-	-	-	-	-
7	@6	LOAD R0	@4	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 4

- El bus de datos pasa **`LOAD R0`** a la CPU
- La CPU pasa **`LOAD R0`** al registro de instrucciones (**RI**)
- La CPU sabe que las instrucciones en la memoria se guardan en 8 bits y **le falta la segunda parte**

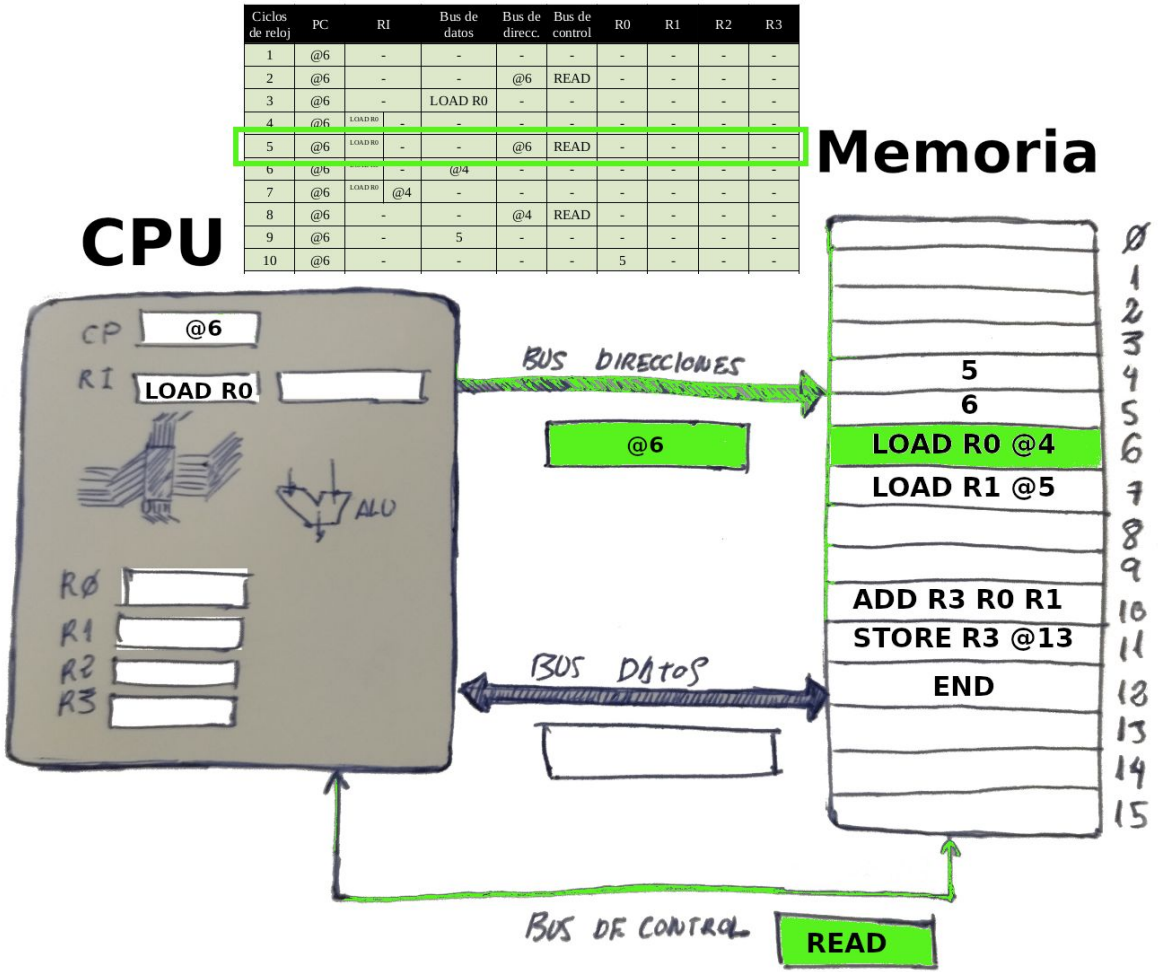
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 5

- La CPU indica al **bus de direcciones** que apunte a `@6`
- La CPU indica al **bus de control** que quiere leer el contenido de la dirección de memoria apuntada (**`READ`**)
- La **instrucción** que se leerá de memoria será: **`LOAD R0 @4`**

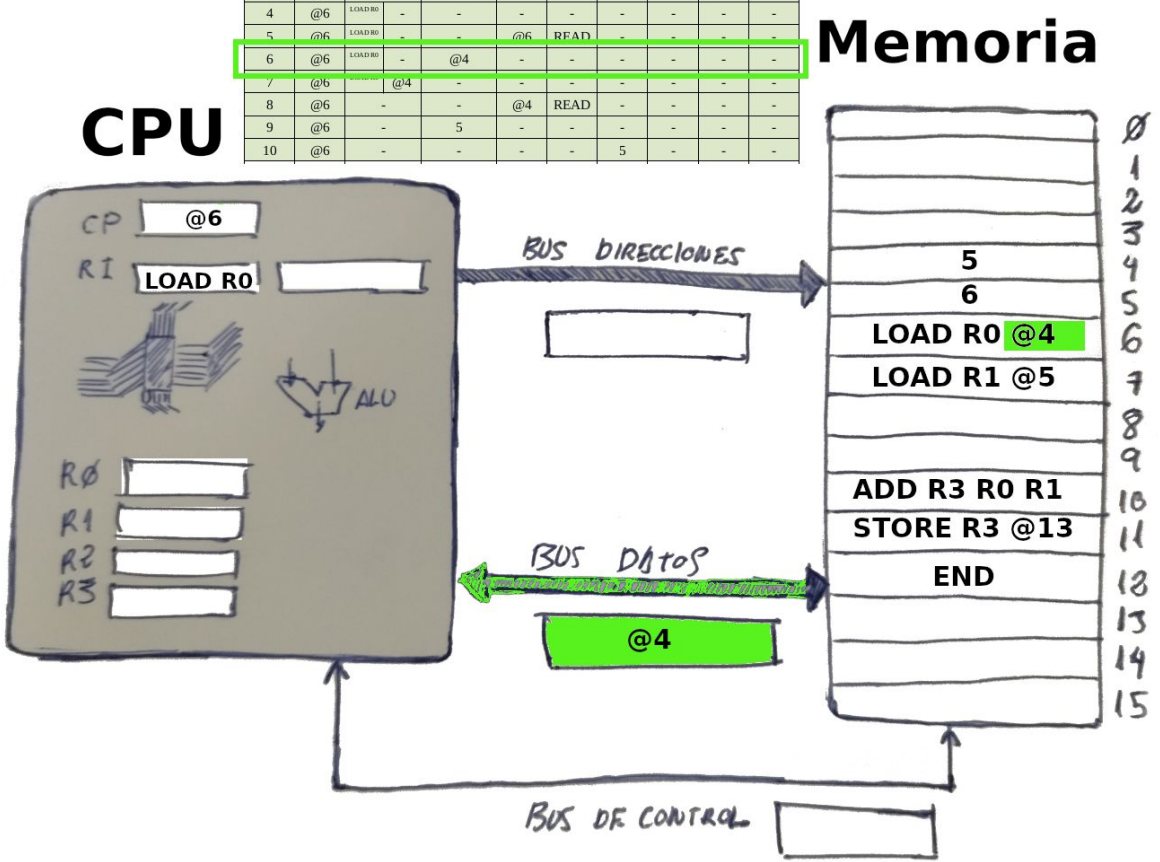
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	-	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 6

- Las instrucciones en la memoria ocupan 8 bits
- La CPU y los buses son solo de 4 bits
- La instrucción se tiene que obtener en 2 partes
- En esta **segunda parte** se obtiene `@4`
- La memoria pasa `@4` al **bus de datos**

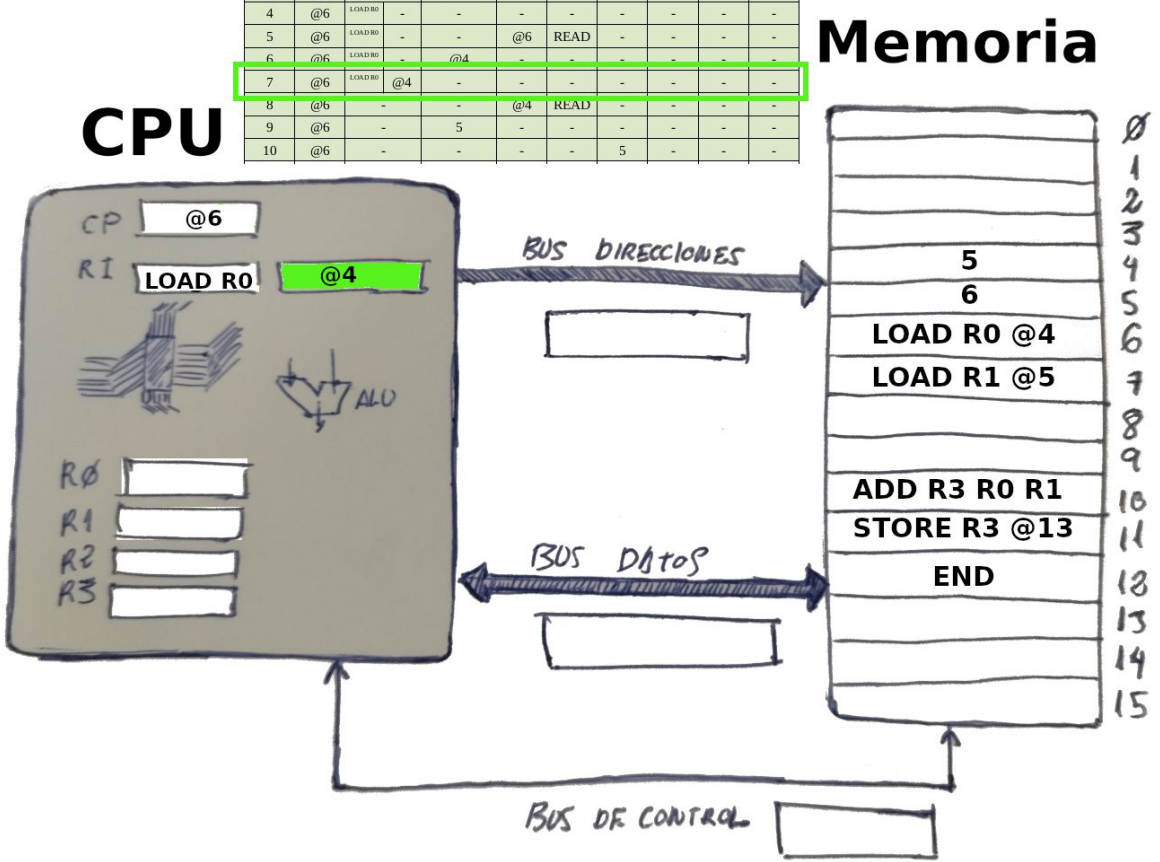
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	@4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 7

- El bus de datos pasa `@4` a la CPU y se almacena en el registro de instrucciones (**RI**)
- La CPU sabe que la instrucción está completa
- La instrucción completa pasa al **decodificador**
- Tras decodificar la instrucción la CPU sabe que debe obtener el dato de la dirección `@4` y almacenarlo en `R0`

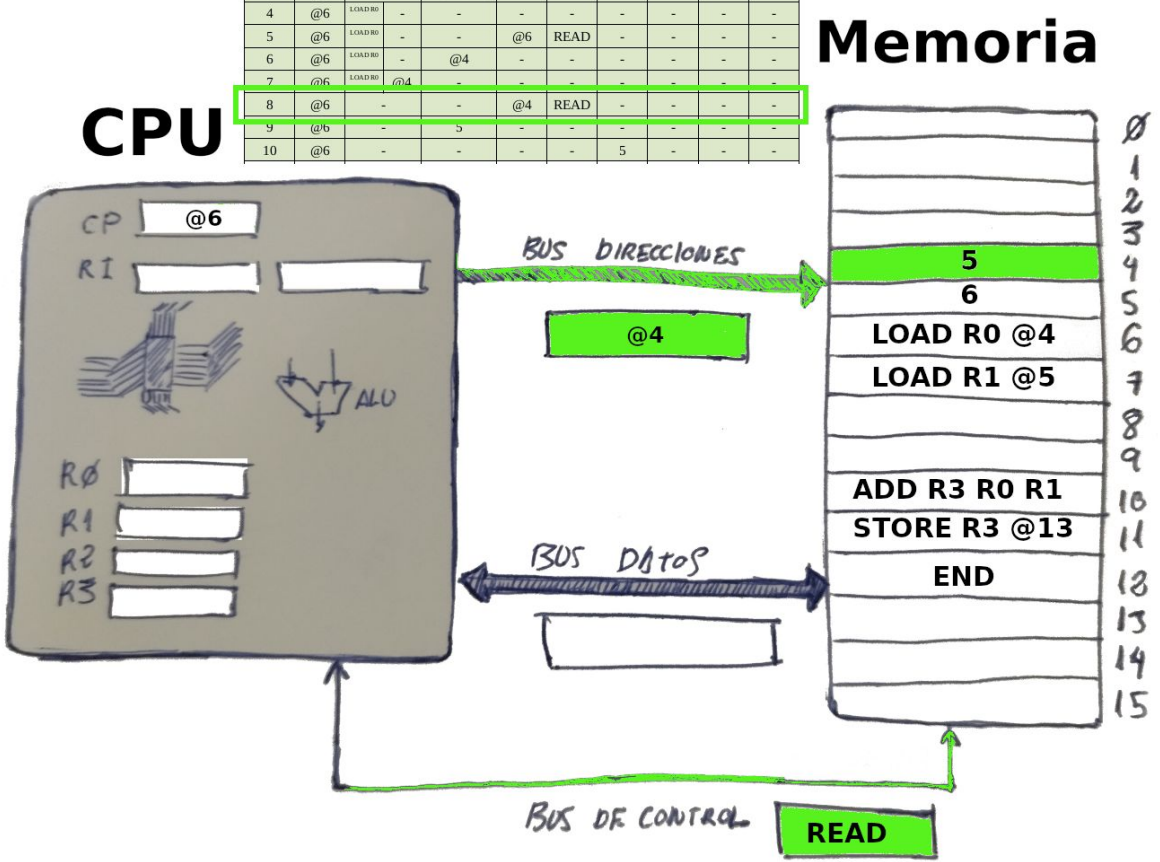
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 8

- La CPU indica al **bus de direcciones** que apunte a **`@4`**
- La CPU indica al **bus de control** que quiere leer el contenido de la dirección de memoria apuntada (**`READ`**)
- El **dato** que se leerá de la memoria será **5**

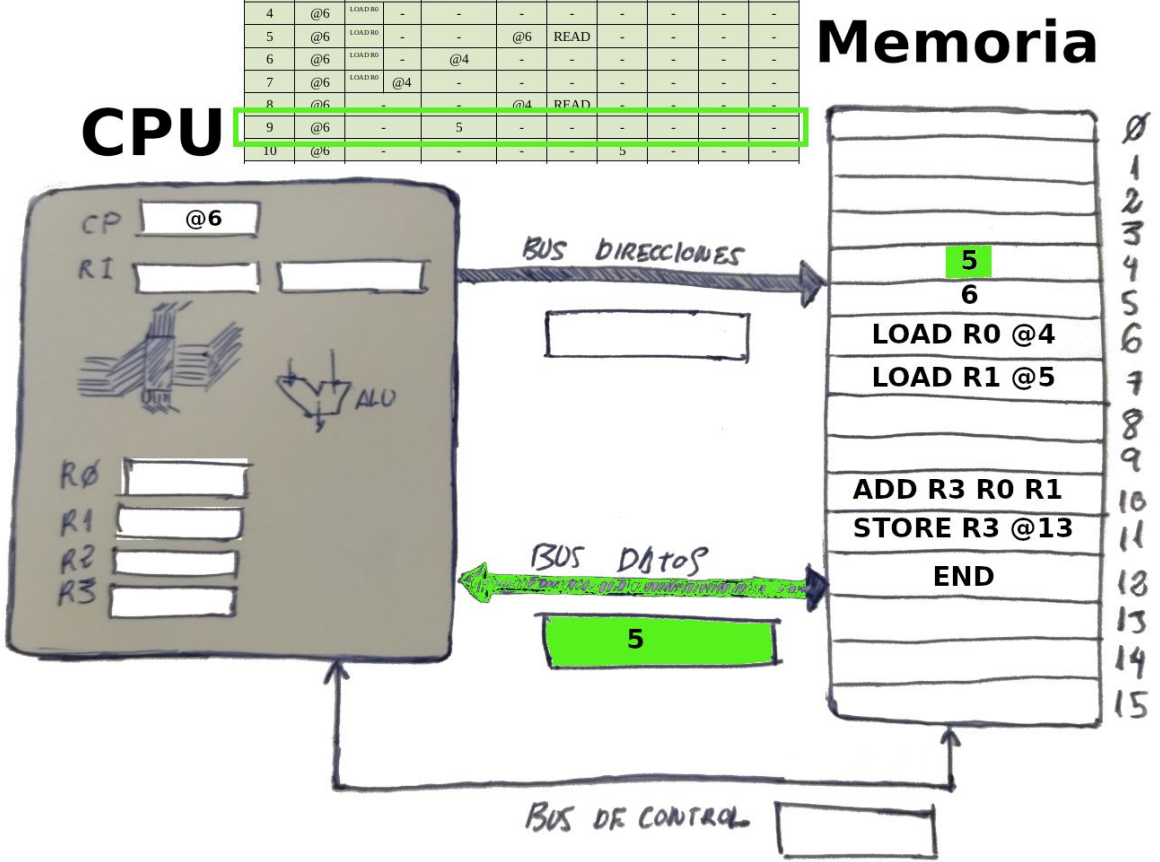
Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 9

- La memoria pasa el dato **5** al **bus de datos**

Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-



Ciclo 10

- La CPU almacena el valor **5** en su registro **R0**

Ciclos de reloj	PC	RI	Bus de datos	Bus de direcc.	Bus de control	R0	R1	R2	R3
1	@6	-	-	-	-	-	-	-	-
2	@6	-	-	@6	READ	-	-	-	-
3	@6	-	LOAD R0	-	-	-	-	-	-
4	@6	LOAD R0 -	-	-	-	-	-	-	-
5	@6	LOAD R0 -	-	@6	READ	-	-	-	-
6	@6	LOAD R0 -	@4	-	-	-	-	-	-
7	@6	LOAD R0 @4	-	-	-	-	-	-	-
8	@6	-	-	@4	READ	-	-	-	-
9	@6	-	5	-	-	-	-	-	-
10	@6	-	-	-	-	5	-	-	-

