

Clase 9: Diseño RTL

Práctico 9 Ejercicio 8

Un sistema digital tiene dos líneas de entrada, **control y data**, 8 líneas de salida **Z[8]**, y una línea de salida **ready**. Los datos representando palabras de 8 bits llegarán serialmente (un bit por vez) sobre la línea data. Será convertido a formato paralelo para sacarlo por las líneas Z. El primer bit en llegar de los datos serie puede ser el más significativo o el menos significativo. En cualquier caso la salida paralelo debe ser de la forma normal, con el más significativo a la izquierda. Se provee sincronización por la interacción entre la entrada control y la salida ready. Cuando el sistema está pronto para recibir una nueva palabra, subirá la línea ready y empezará a monitorear control. Dos unos consecutivos (igual a uno durante dos períodos de reloj) en control indicará que el siguiente bit en control indicará la dirección de los datos serie, 0 para el MSB primero, 1 para el LSB primero. Los 8 bits de datos aparecerán a continuación en data durante los siguientes 8 períodos de reloj. Ready deberá bajar cuando los dos unos consecutivos en control se detecten y deberá volver a 1 cuando los 8 bits hayan sido recibidos. La palabra paralelo deberá presentarse en las líneas Z siempre que ready esté alta.

Descripción RTL

MODULE: Ej_8

MEMORY:

INPUTS: control, data

OUTPUTS: Z[8], ready

Encabezado: nombre, entradas, salidas, memoria

1.

Pasos secuenciales

ENDSEQUENCE

Las sentencias luego del ENSEQUENCE son válidas para todos los pasos

CONTROLRESET (1)

A qué paso voy luego de un reset

END

Descripción RTL

0 para el MSB primero, 1 para el LSB primero

MODULE: Ej_8

MEMORY: palabra[8], cont[3]

INPUTS: control, data

OUTPUTS: Z[8], ready

1. ready=1, Z=palabra, \rightarrow (control=1,! (control=1))/(2,1)
2. ready=1, Z=palabra, \rightarrow (control=1,! (control=1))/(3,1)
3. ready=0, cont \leftarrow 0, \rightarrow (control=1,! (control=1))/(4,5)
4. ready=0, palabra \leftarrow data,palabra[7..1], cont \leftarrow cont +1, \rightarrow (cont=7,! (cont=7))/(1,4)
5. ready=0, palabra \leftarrow palabra[6..0],data, cont \leftarrow cont +1, \rightarrow (cont=7,! (cont=7))/(1,5)

ENDSEQUENCE

CONTROLRESET (1)

END

Descripción RTL

MODULE: Ej_8

MEMORY: palabra[8], cont[3], sentido

INPUTS: control, data

OUTPUTS: Z[8], ready

1. $ready=1, \rightarrow (control=1,!(control=1))/(2,1)$
2. $ready=1, \rightarrow (control=1,!(control=1))/(3,1)$
3. $cont \leftarrow 0, sentido \leftarrow control$
4. $palabra \leftarrow (data,palabra[7..1] \wedge sentido) \vee (palabra[6..0],data \wedge !sentido),$
 $cont \leftarrow cont + 1,$
 $\rightarrow (cont=7,!(cont=7))/(1,4)$

ENDSEQUENCE

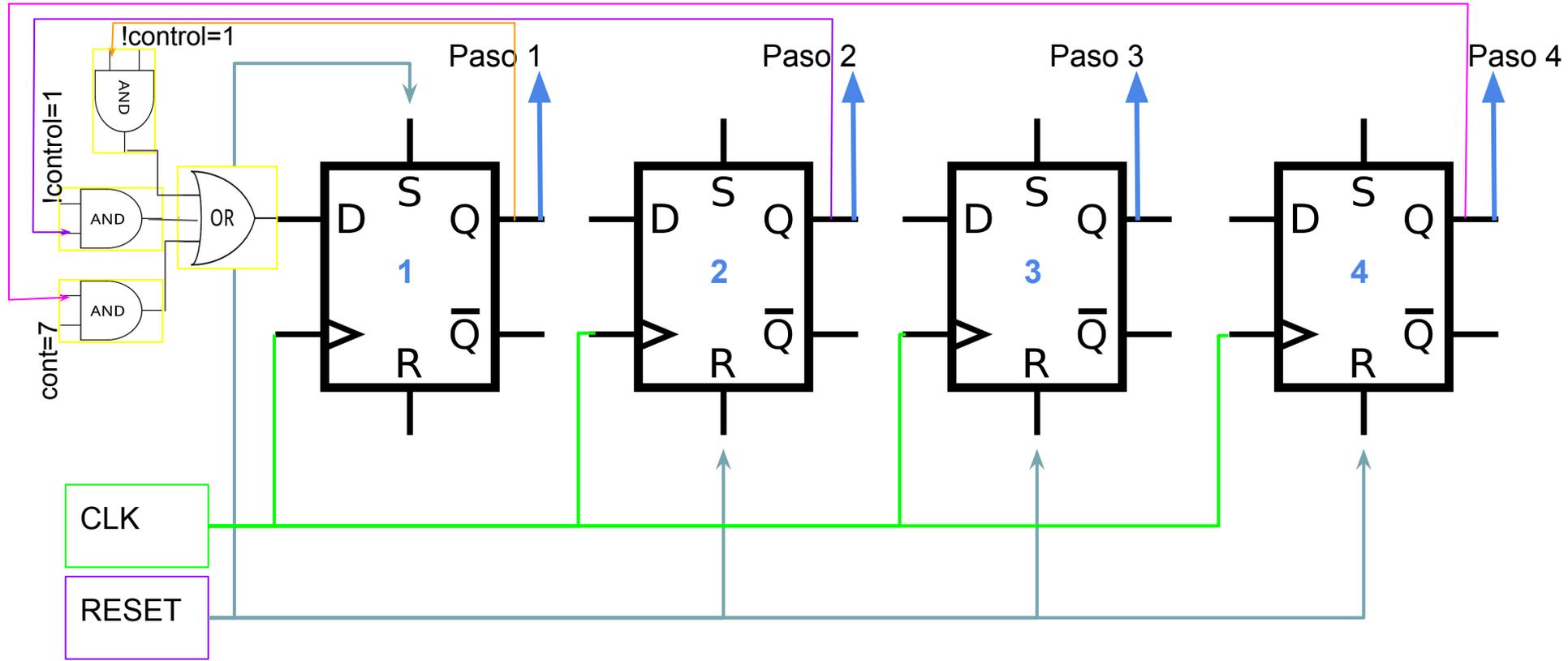
Z=palabra

CONTROLRESET (1)

END

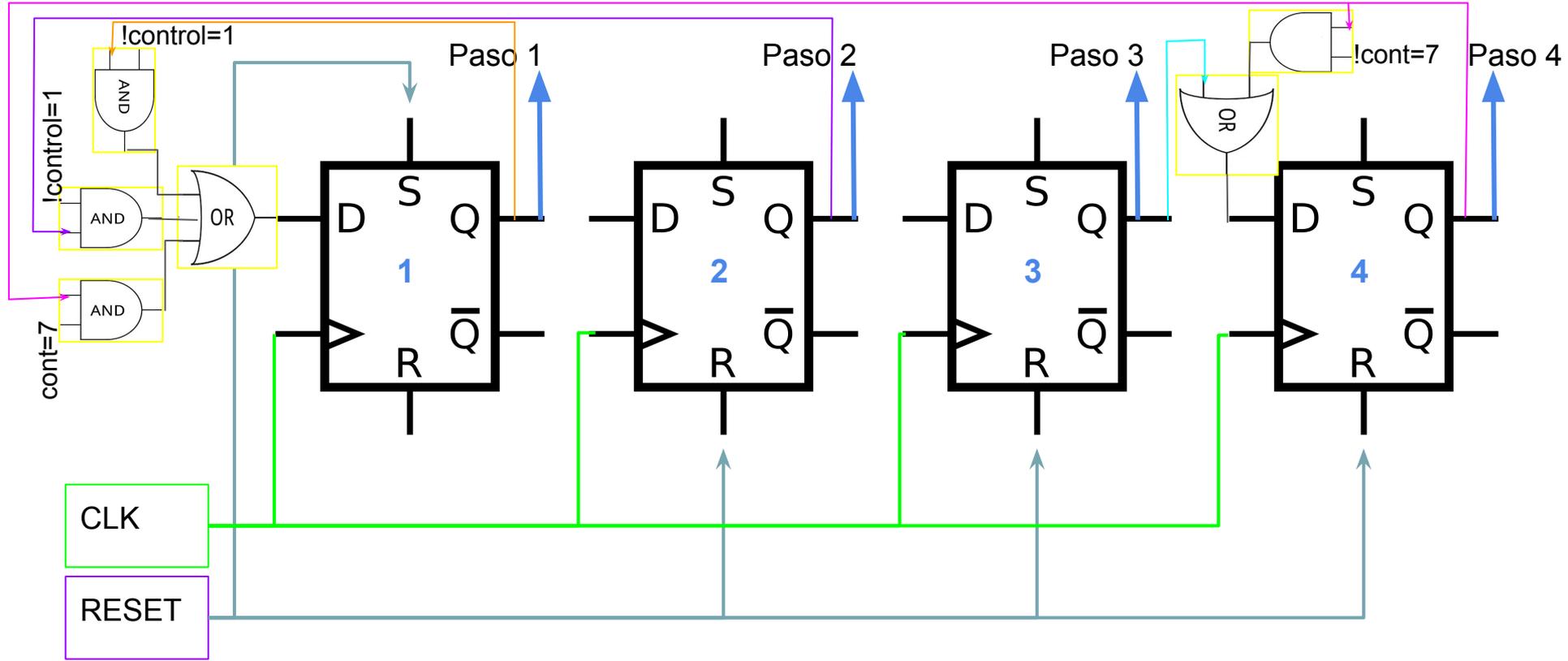
Bloque de Control

1. $\rightarrow (\text{control}=1,!(\text{control}=1))/(2,1)$
 2. $\rightarrow (\text{control}=1,!(\text{control}=1))/(3,1)$
 - 3.
 4. $\rightarrow (\text{cont}=7,!(\text{cont}=7))/(1,4)$
- CONTROLRESET (1)



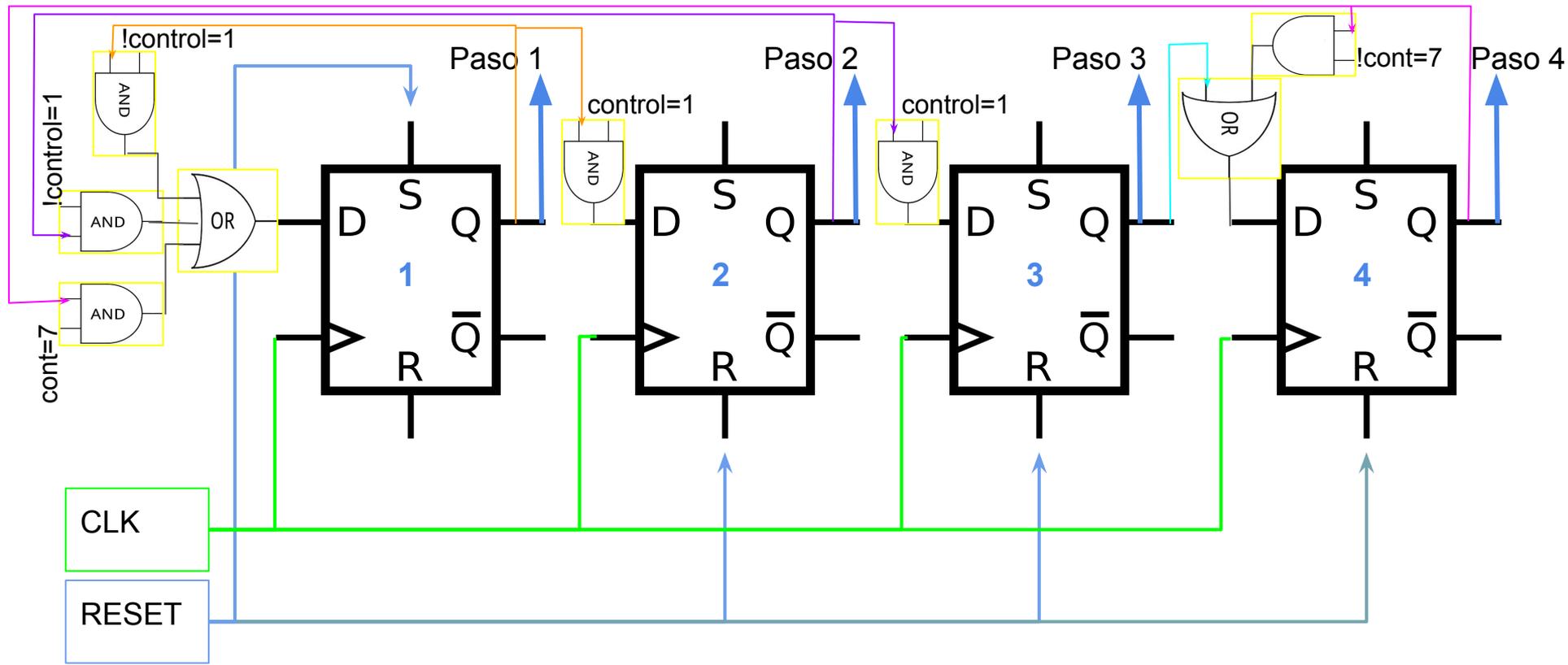
Bloque de Control

1. $\rightarrow (\text{control}=1,!(\text{control}=1))/(2,1)$
 2. $\rightarrow (\text{control}=1,!(\text{control}=1))/(3,1)$
 - 3.
 4. $\rightarrow (\text{cont}=7,!(\text{cont}=7))/(1,4)$
- CONTROLRESET (1)

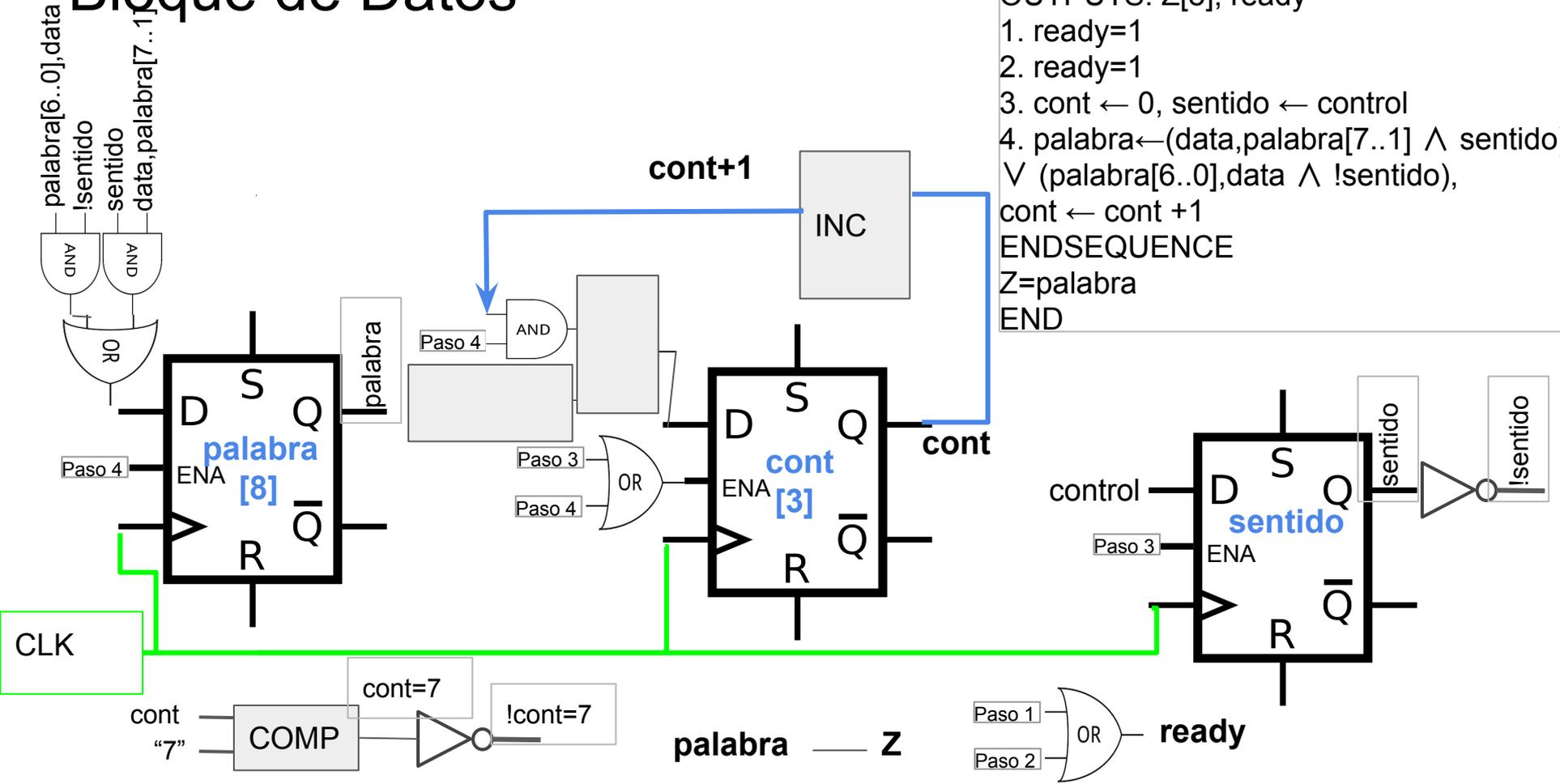


Bloque de Control

1. $\rightarrow (\text{control}=1,!(\text{control}=1))/(2,1)$
 2. $\rightarrow (\text{control}=1,!(\text{control}=1))/(3,1)$
 - 3.
 4. $\rightarrow (\text{cont}=7,!(\text{cont}=7))/(1,4)$
- CONTROLRESET (1)



Bloque de Datos



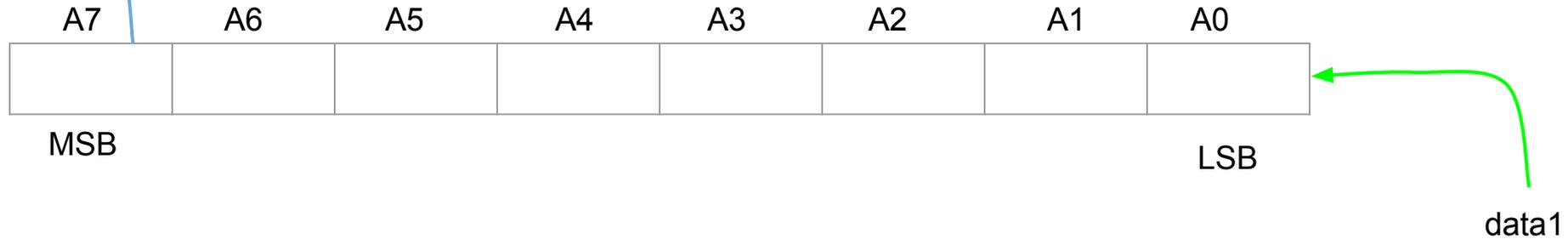
MEMORY: palabra[8], cont[3], sentido
 INPUTS: control, data
 OUTPUTS: Z[8], ready

- ready=1
- ready=1
- cont ← 0, sentido ← control
- palabra ← (data, palabra[7..1] ∧ sentido) ∨ (palabra[6..0], data ∧ !sentido),
 cont ← cont + 1

ENDSEQUENCE
 Z=palabra
 END

sentido de palabra

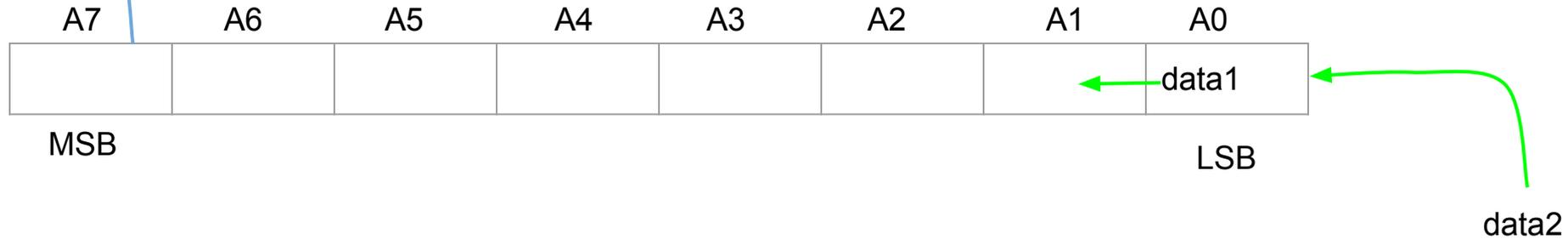
sentido = 0 para el MSB primero



cont=0

sentido de palabra

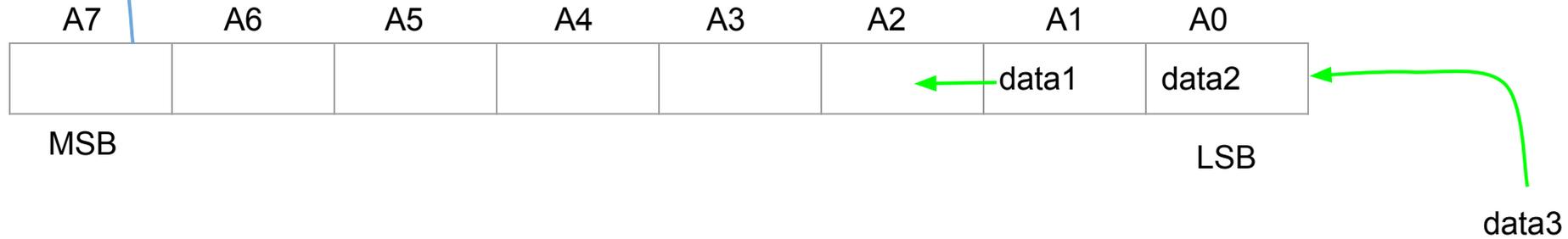
sentido = 0 para el MSB primero



cont=1

sentido de palabra

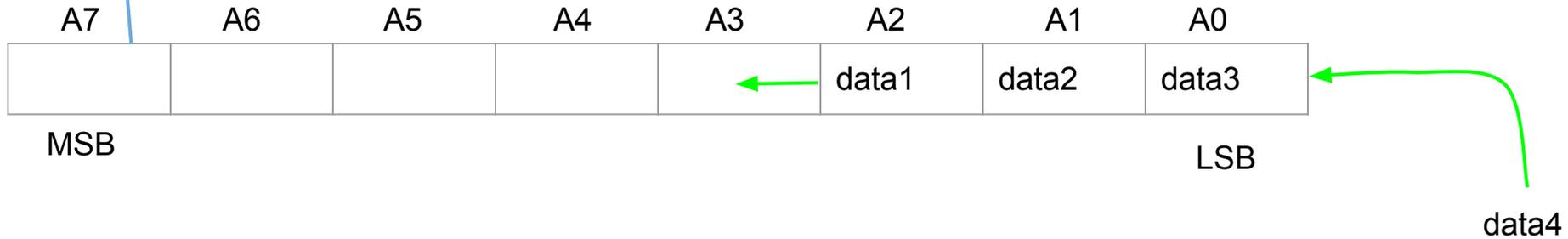
sentido = 0 para el MSB primero



cont=2

sentido de palabra

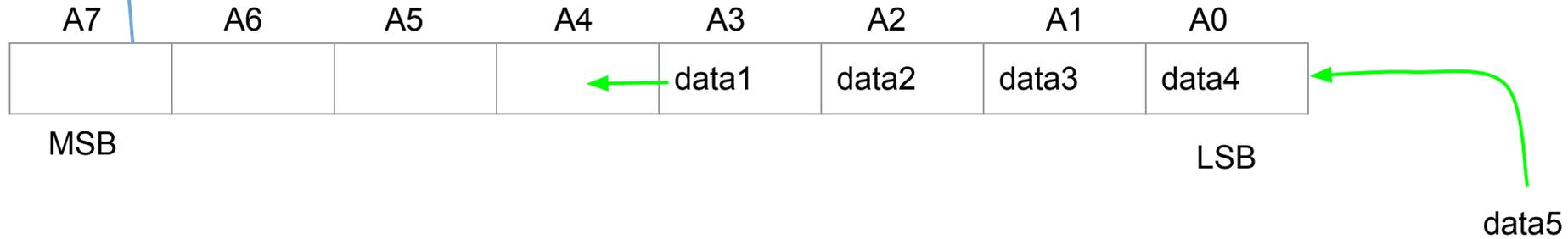
sentido = 0 para el MSB primero



cont=3

sentido de palabra

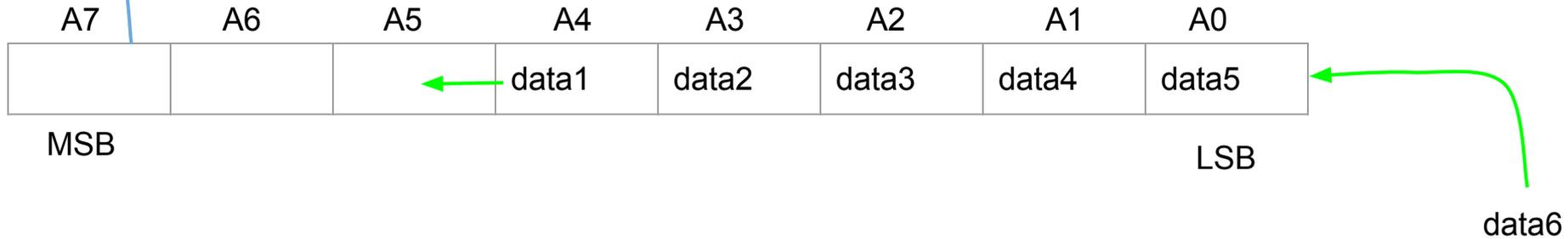
sentido = 0 para el MSB primero



cont=4

sentido de palabra

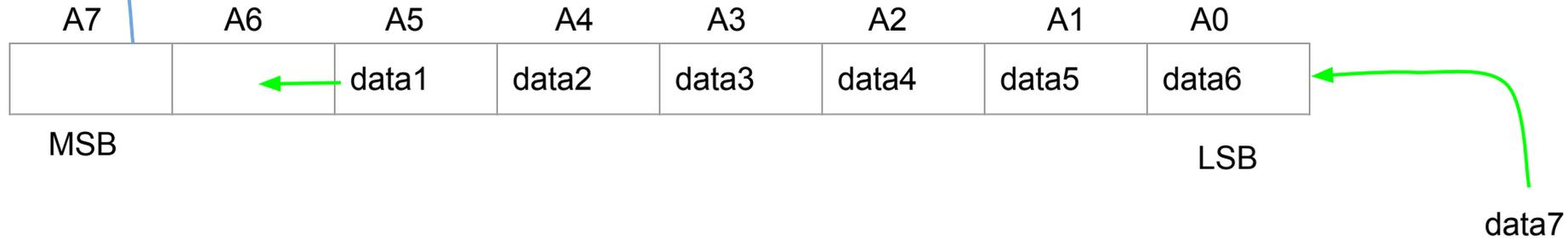
sentido = 0 para el MSB primero



cont=5

sentido de palabra

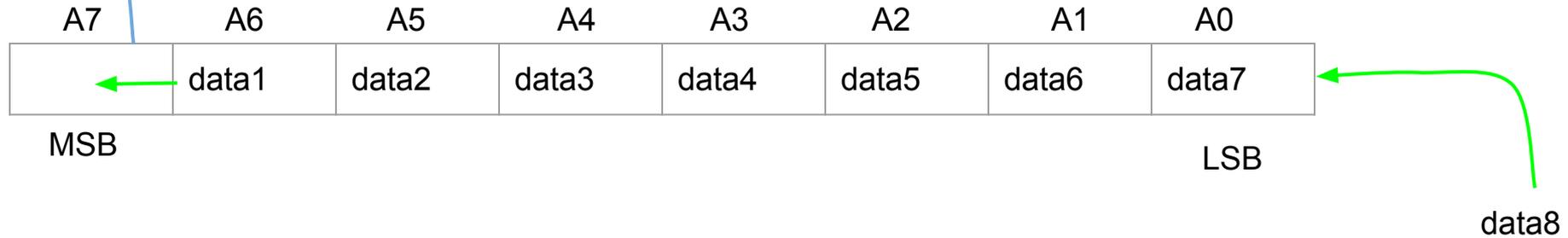
sentido = 0 para el MSB primero



cont=6

sentido de palabra

sentido = 0 para el MSB primero



cont=7

sentido de palabra

sentido = 0 para el MSB primero

