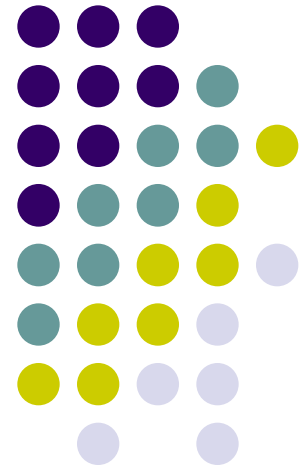


# Práctica Laboratorio

## TP 3

### Arquitectura de Computadoras

Prof. Dr. Martín Vázquez



# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- *MemoryCache.vhd*
- *ProcesssorCacheTB.vhd*

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- *program3.s*
- *PseudoCódigo3.pdf*

# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*

- *MemoryCache.vhd*

- *ProcessorCacheTB.vhd*

- Memoria Principal (RAM)

- Se lee y escribe bloques

- Bloques de 8 palabras de 4 bytes (o 32 bytes)



- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*


- *program3*

- *program3.s*

- *PseudoCódigo3.pdf*

# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- ***MemoryCache.vhd*** 
- *ProcessorCacheTB.vhd*

- **Memoria Cache**

- **Mapeo directo de 4 vías o líneas**

- **O asociativa con 4 conjuntos de una línea**


- **Una línea puede ubicar un bloque de 8 palabras de 4 bytes**

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- *program3.s*
- *PseudoCódigo3.pdf*

# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- *MemoryCache.vhd*
- ***ProcessorCacheTB.vhd*** 

- ***TestBench.***

- **Instancia el procesador y las tres memorias.**

- **Solo se modificará en el caso que se requiera cambiar el contenido de las memorias**

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- *program3.s*
- *PseudoCódigo3.pdf*

# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- *MemoryCache.vhd*
- *PrcessorCacheTB.vhd*

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- *program3.s*
- *PseudoCódigo3.pdf*



- Archivos de texto con el contenido de las memorias de datos y programa.
- Las memorias se cargan con esos archivos

# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- *MemoryCache.vhd*
- *ProcessorCacheTB.vhd*

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- ***program3.s***
- *PseudoCódigo3.pdf*

- Archivo con código assembler del programa cargado en memoria.



# Archivos que proporciona la cátedra

- **Archivos VHDL**

- *DataMemoryBlockPrincipal.vhd*
- *MemoryCache.vhd*
- *ProcessorCacheTB.vhd*

- **Archivos contenidos de memorias (Datos y Programa)**

- *data3*
- *program3*
- *program3.s*

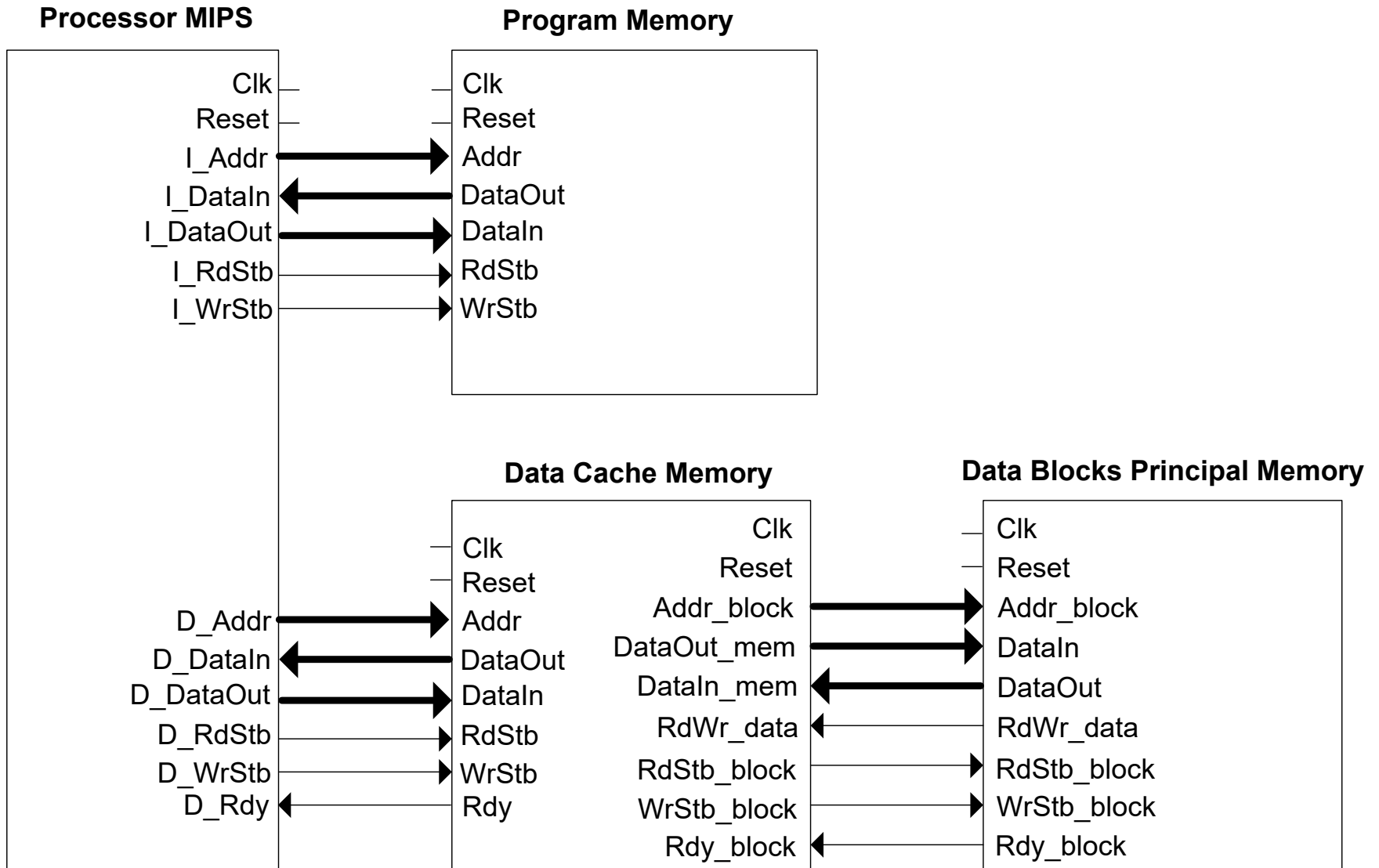
- ***PseudoCódigo3.pdf***

- Descripción abstracta del programa 3 (PseudoCódigo)

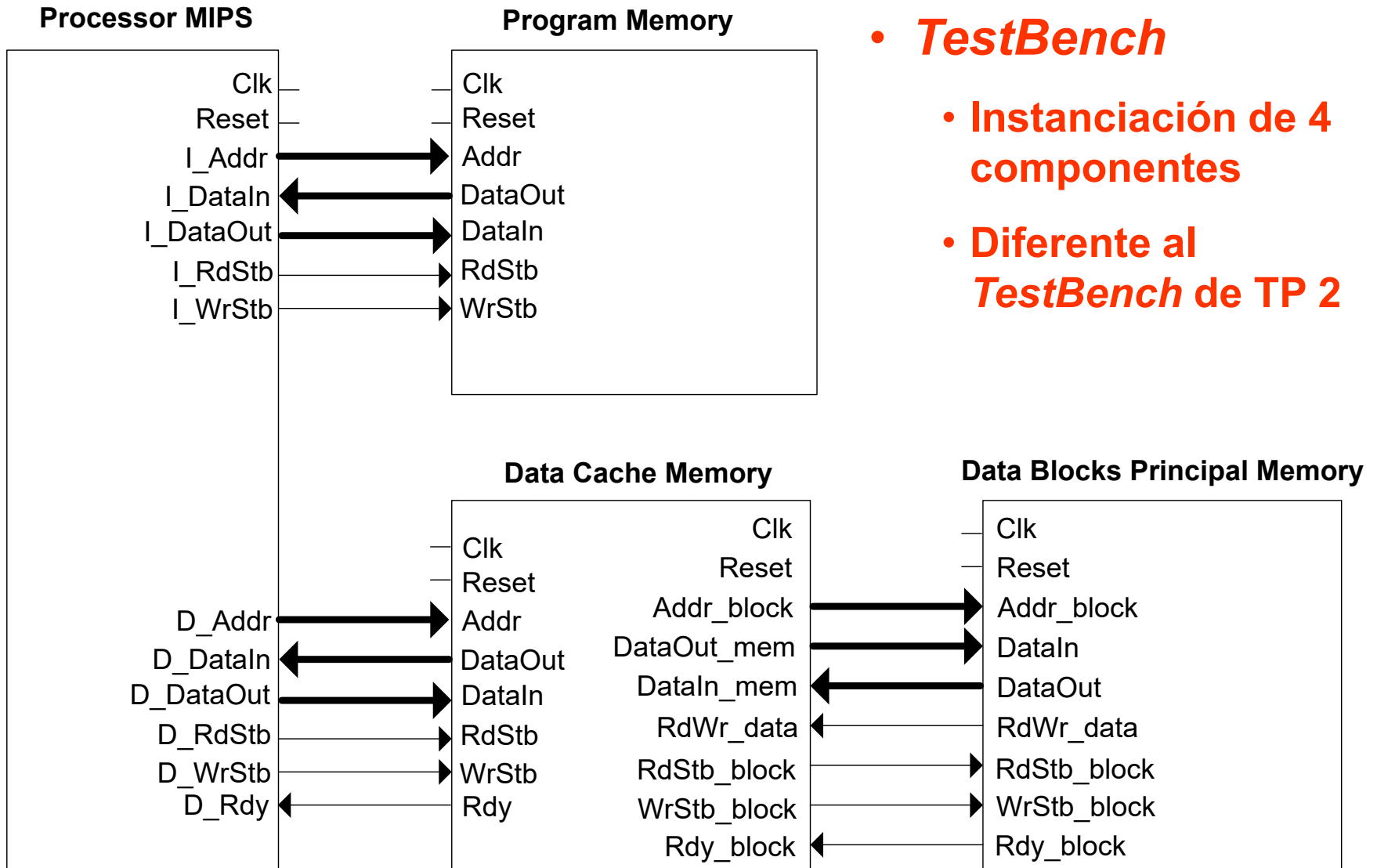




# Diagrama de bloques del TP 3



# Diagrama de bloques del TP 3

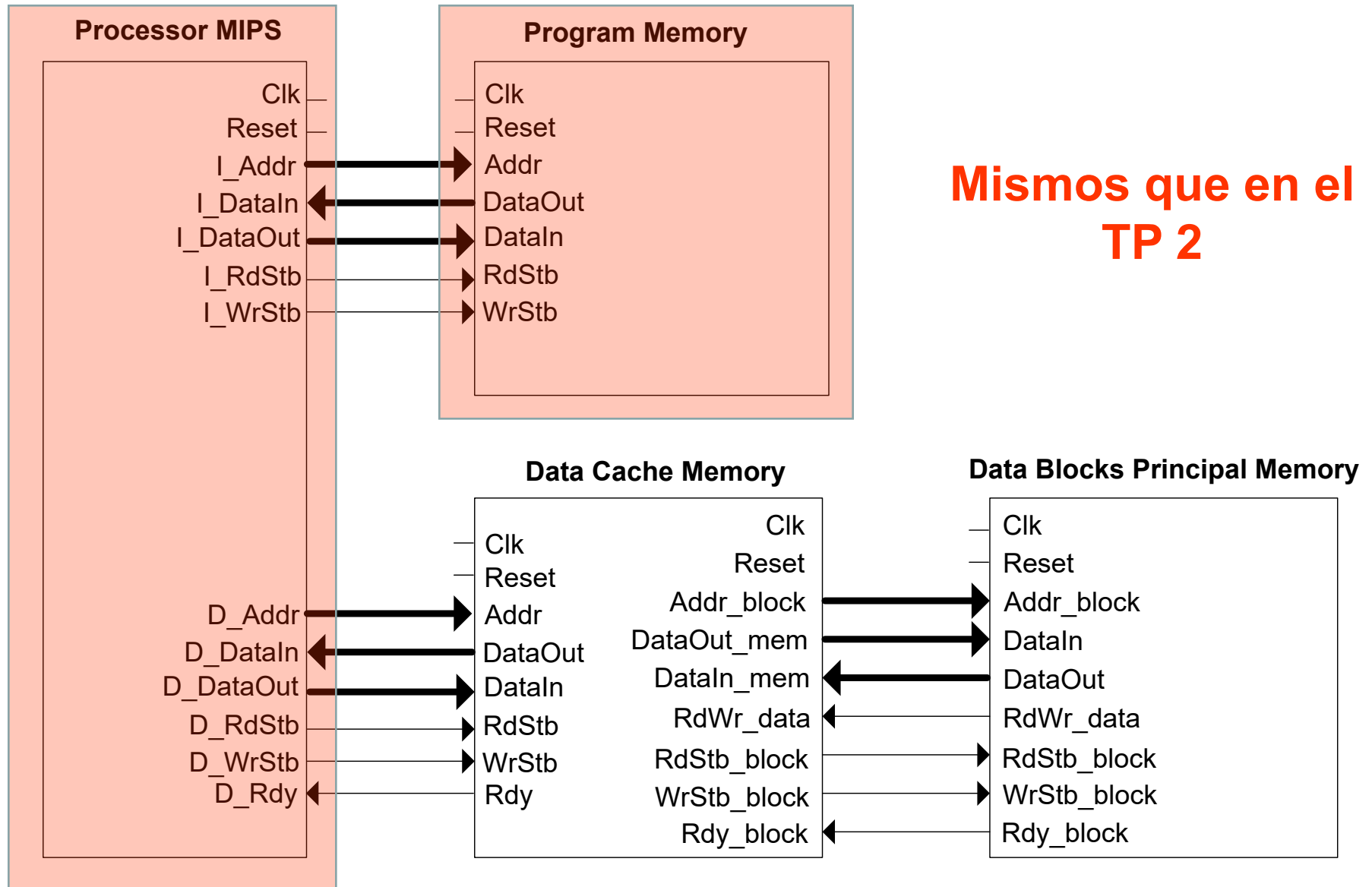


- ***TestBench***

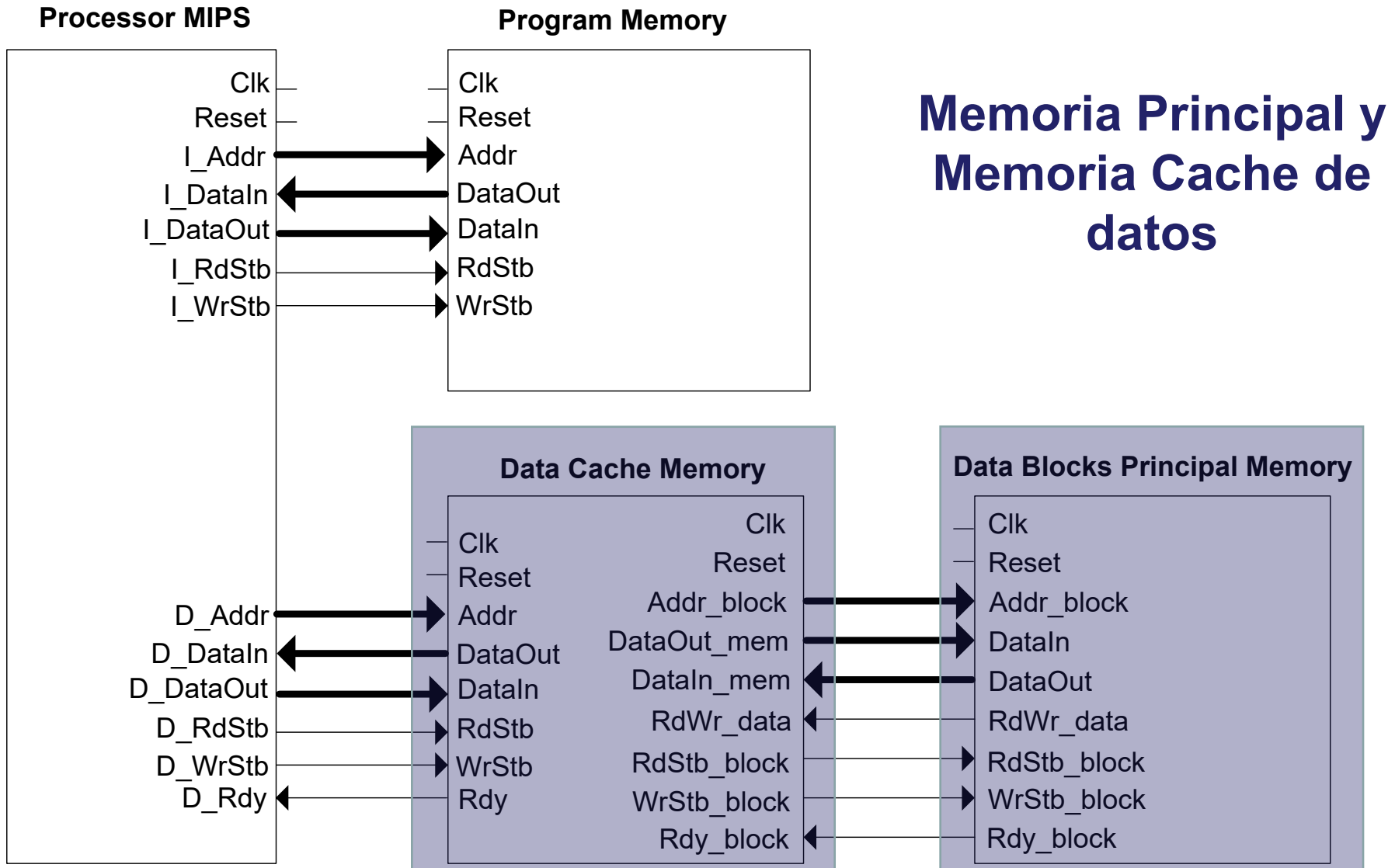
- **Instanciación de 4 componentes**

- **Diferente al *TestBench* de TP 2**

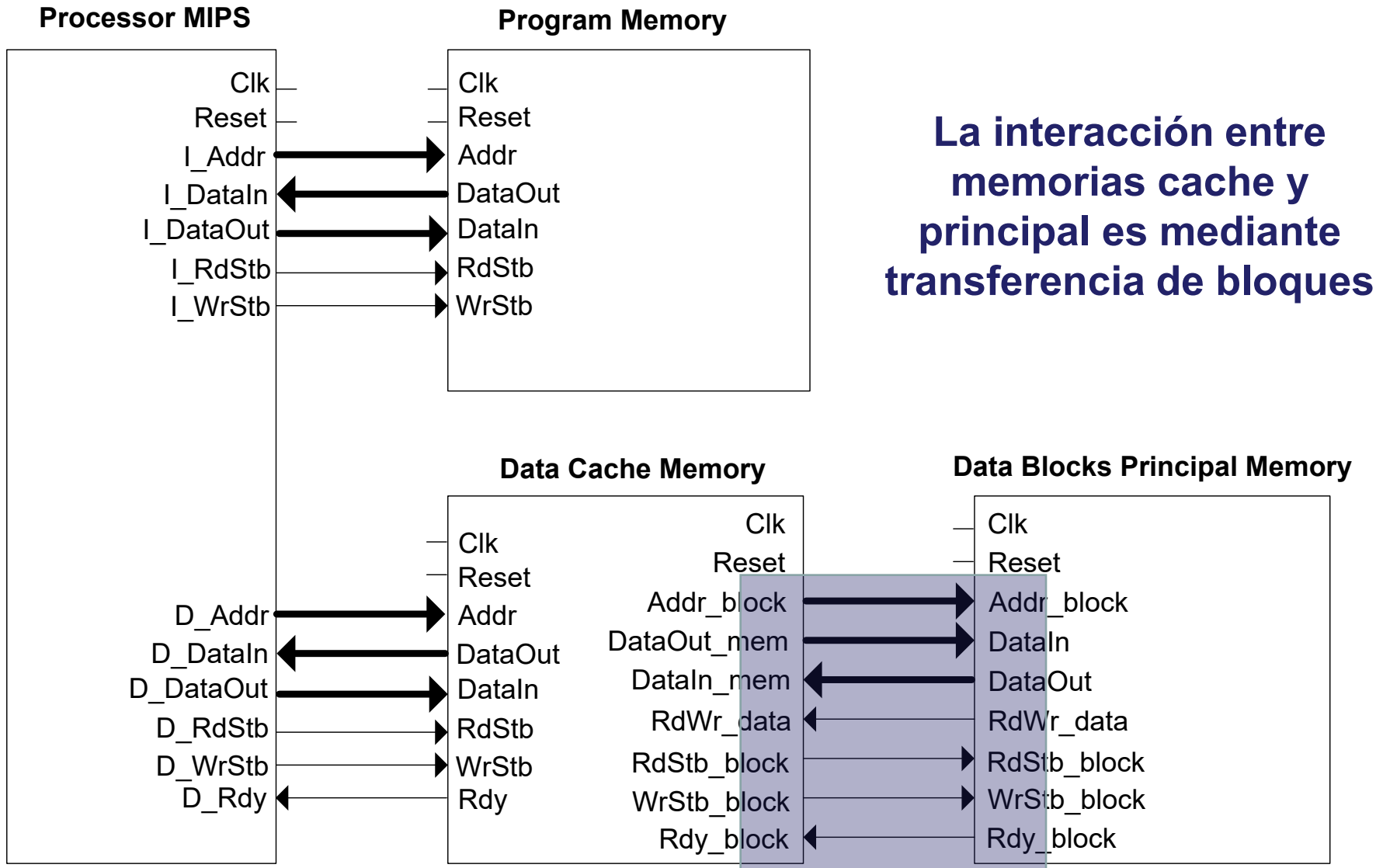
# Diagrama de bloques del TP 3



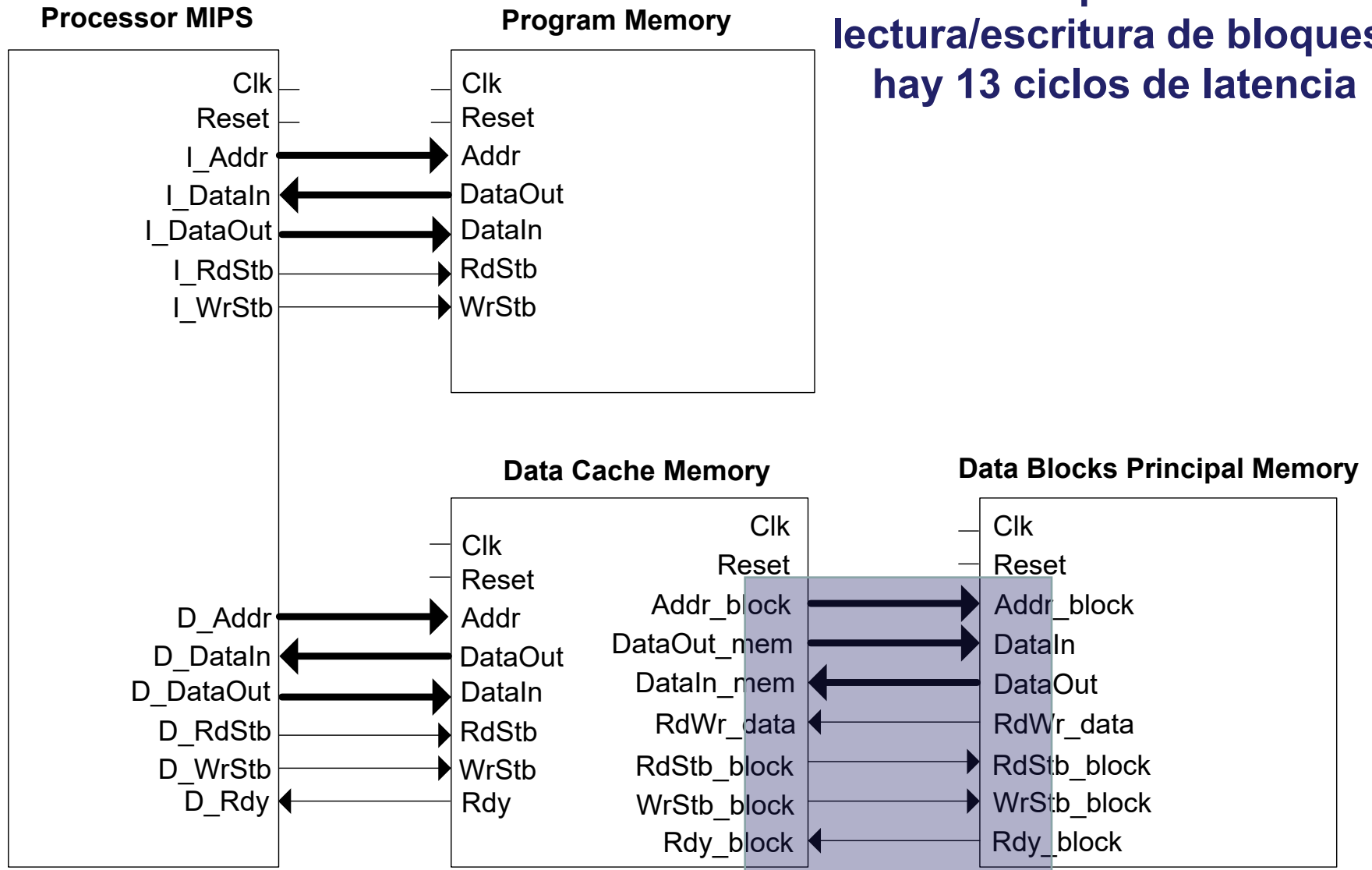
# Diagrama de bloques del TP 3



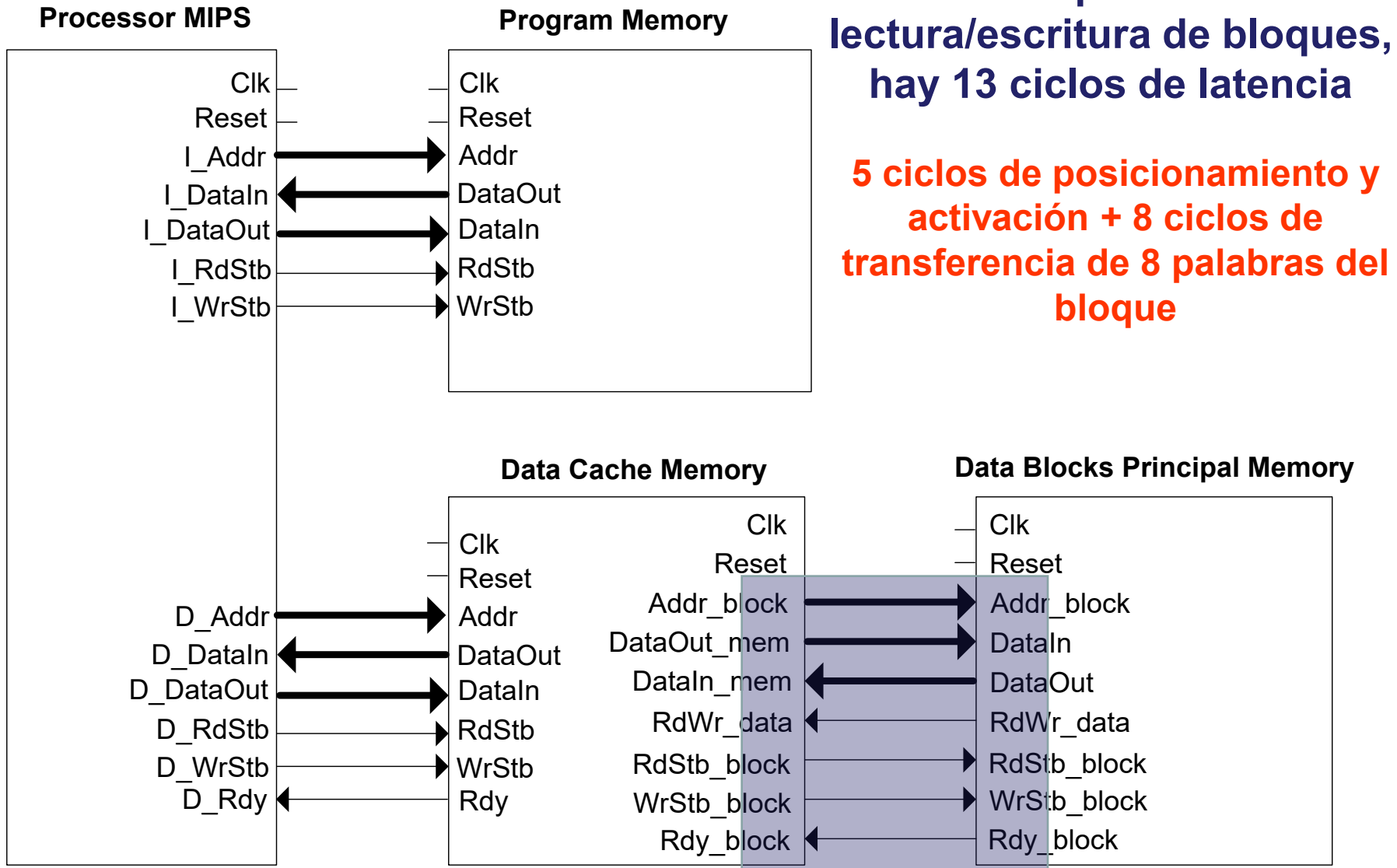
# Diagrama de bloques del TP 3



# Diagrama de bloques del TP 3



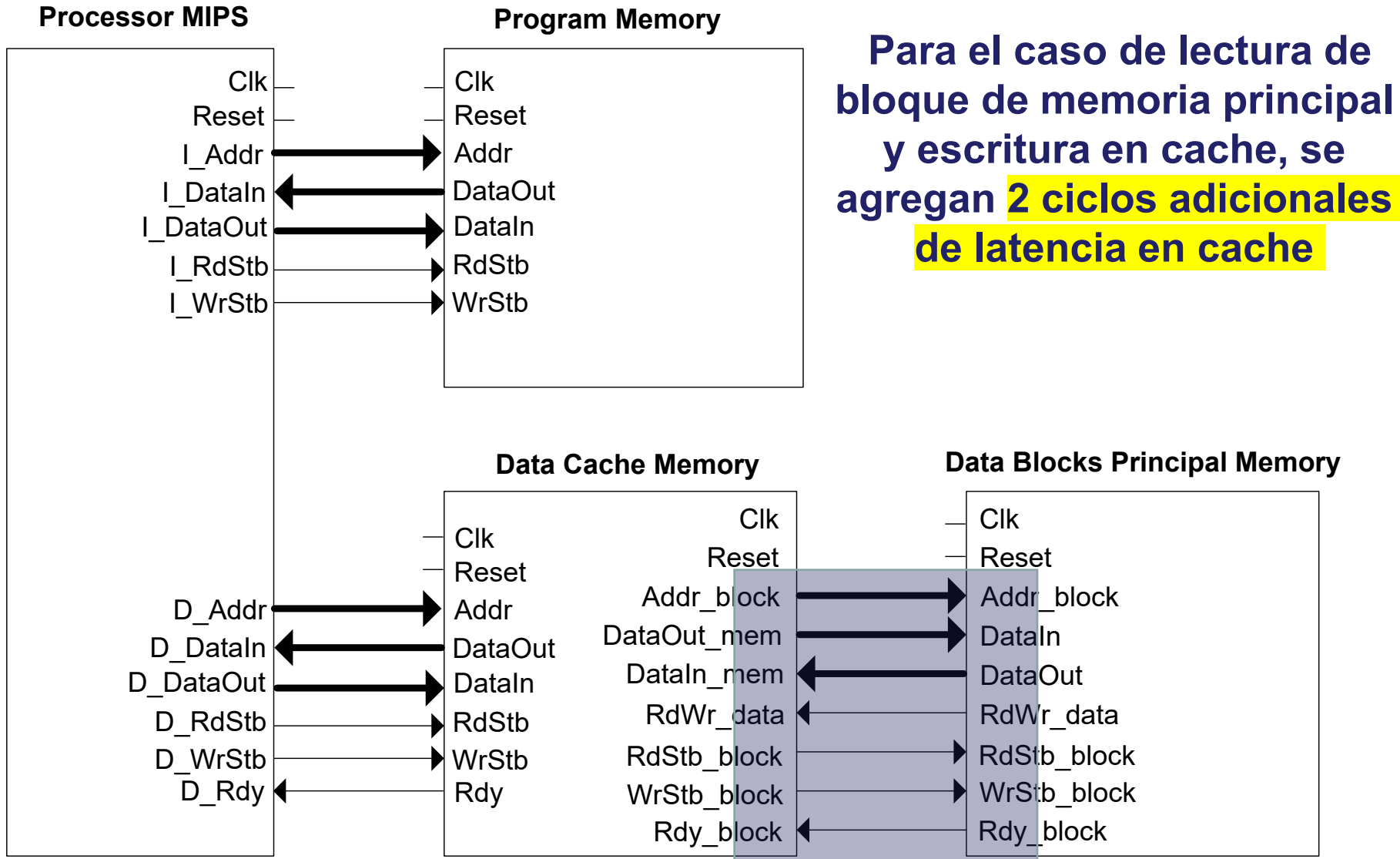
# Diagrama de bloques del TP 3



Para las operaciones de lectura/escritura de bloques, hay 13 ciclos de latencia

5 ciclos de posicionamiento y activación + 8 ciclos de transferencia de 8 palabras del bloque

# Diagrama de bloques del TP 3





# Diagrama de bloques del TP 3

## Processor MIPS

Clk  
Reset  
I\_Addr  
I\_DataIn  
I\_DataOut  
I\_RdStb  
I\_WrStb

**Addr\_block:** dirección de bloque a leer o escribir  
**RdStb\_block:** indica si requiere lectura de bloque  
**WrStb\_block:** indica si requiere escritura de bloque  
**Rdy\_block:** indica que la memoria principal está disponible

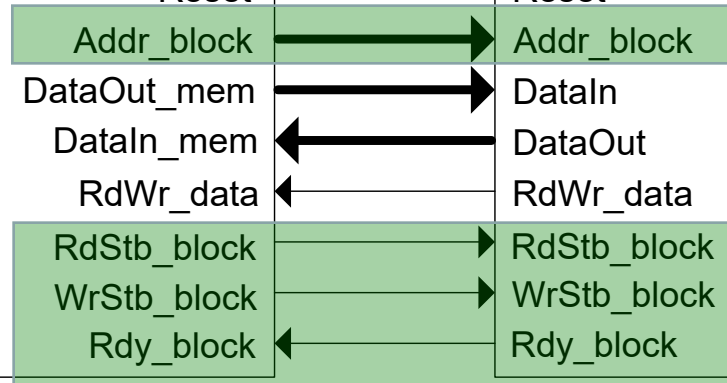
## Data Cache Memory

D\_Addr  
D\_DataIn  
D\_DataOut  
D\_RdStb  
D\_WrStb  
D\_Rdy

Clk  
Reset  
Addr  
DataOut  
DataIn  
RdStb  
WrStb  
Rdy

## Data Blocks Principal Memory

Clk  
Reset  
Addr\_block  
DataIn  
DataOut  
RdWr\_data  
RdStb\_block  
WrStb\_block  
Rdy\_block



# Diagrama de bloques del TP 3

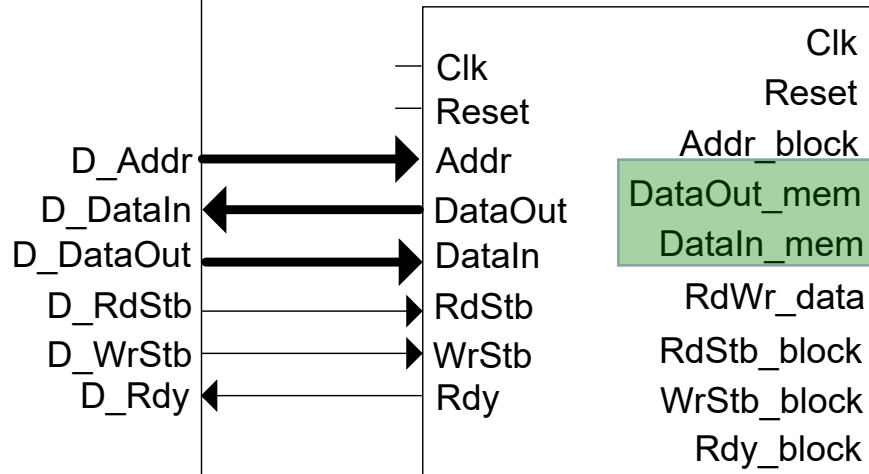
## Processor MIPS

Clk  
Reset  
I\_Addr  
I\_DataIn  
I\_DataOut  
I\_RdStb  
I\_WrStb

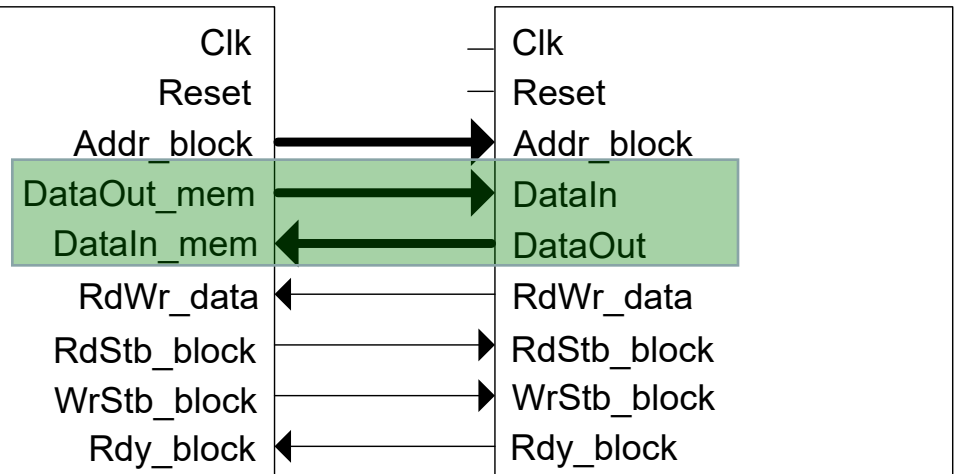
**DataIn:** palabra del bloque direccionado por la cache que entra a memoria principal para escritura

**DataOut:** palabra leída del bloque de memoria principal direccionado por la cache

## Data Cache Memory



## Data Blocks Principal Memory



# Diagrama de bloques del TP 3

## Processor MIPS

Clk  
Reset  
I\_Addr  
I\_DataIn  
I\_DataOut  
I\_RdStb  
I\_WrStb

**RdWr\_data:** indica a memoria cache, que la memoria principal está lista para comenzar a leer o escribir de manera serial los ocho datos del bloque direccionado

## Data Cache Memory

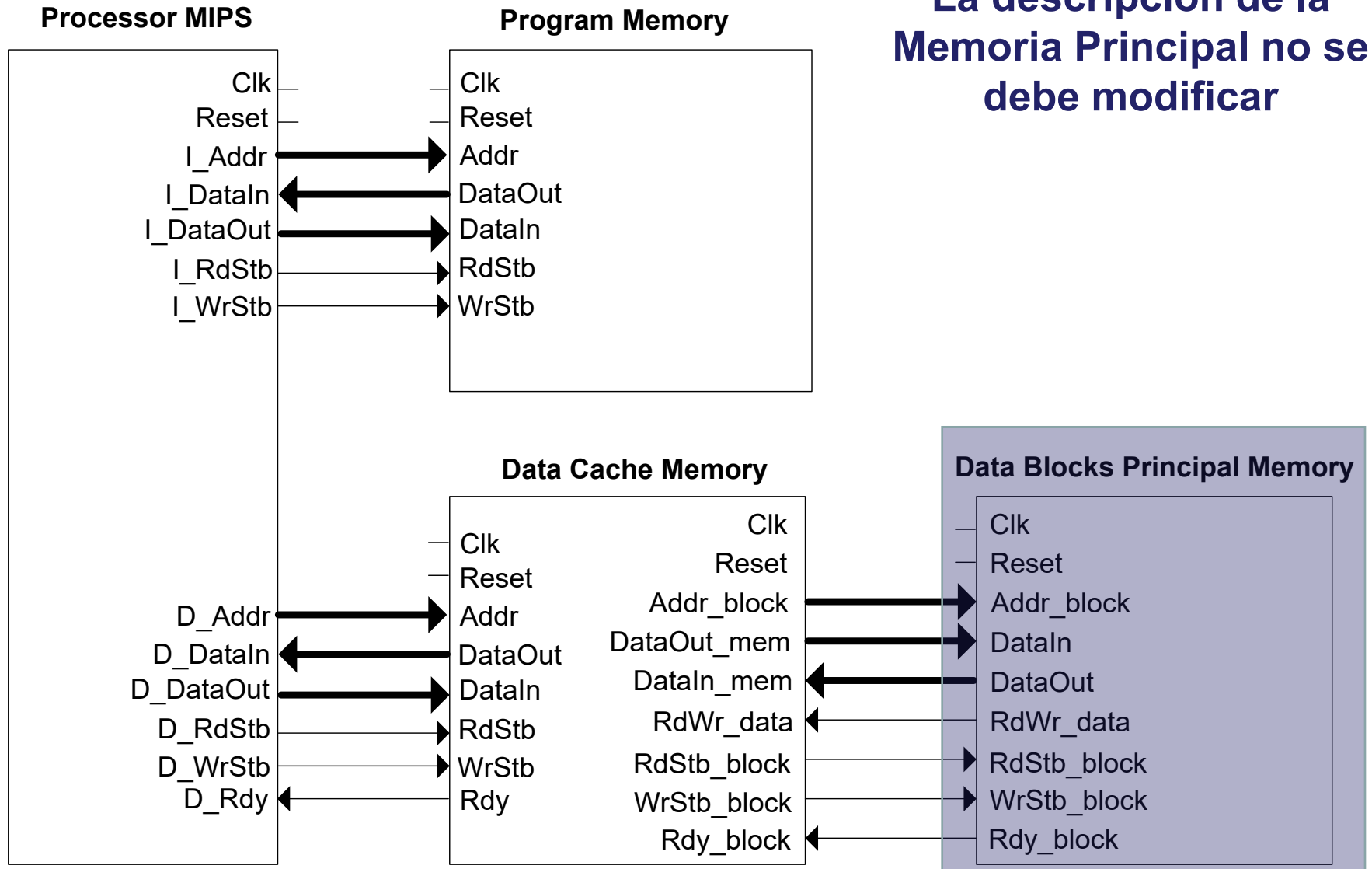
Clk  
Reset  
D\_Addr  
D\_DataIn  
D\_DataOut  
D\_RdStb  
D\_WrStb  
D\_Rdy

Clk  
Reset  
Addr\_block  
DataOut\_mem  
DataIn\_mem  
RdWr\_data  
RdStb\_block  
WrStb\_block  
Rdy\_block

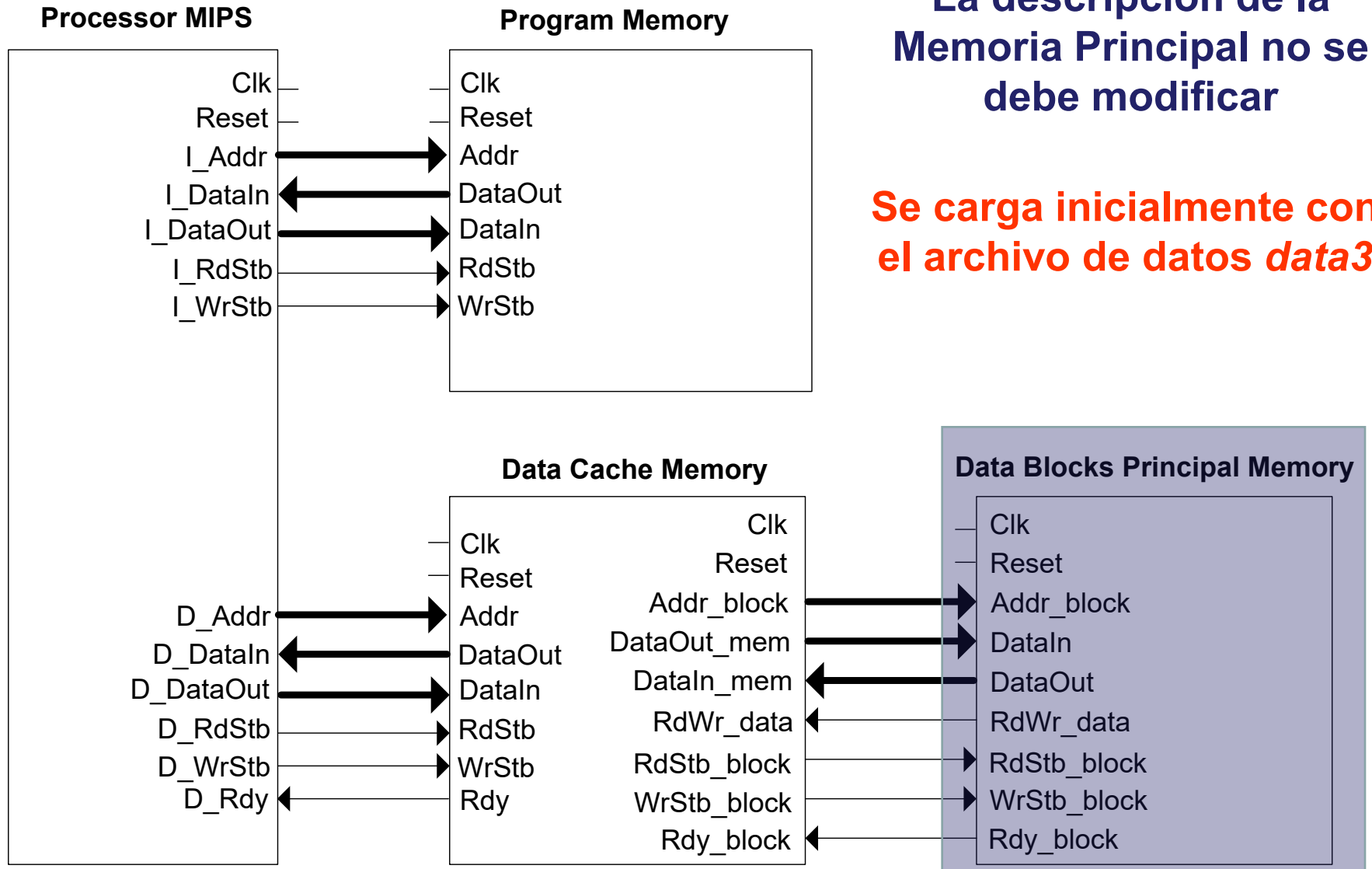
## Data Blocks Principal Memory

Clk  
Reset  
Addr\_block  
DataIn  
DataOut  
RdWr\_data  
RdStb\_block  
WrStb\_block  
Rdy\_block

# Diagrama de bloques del TP 3



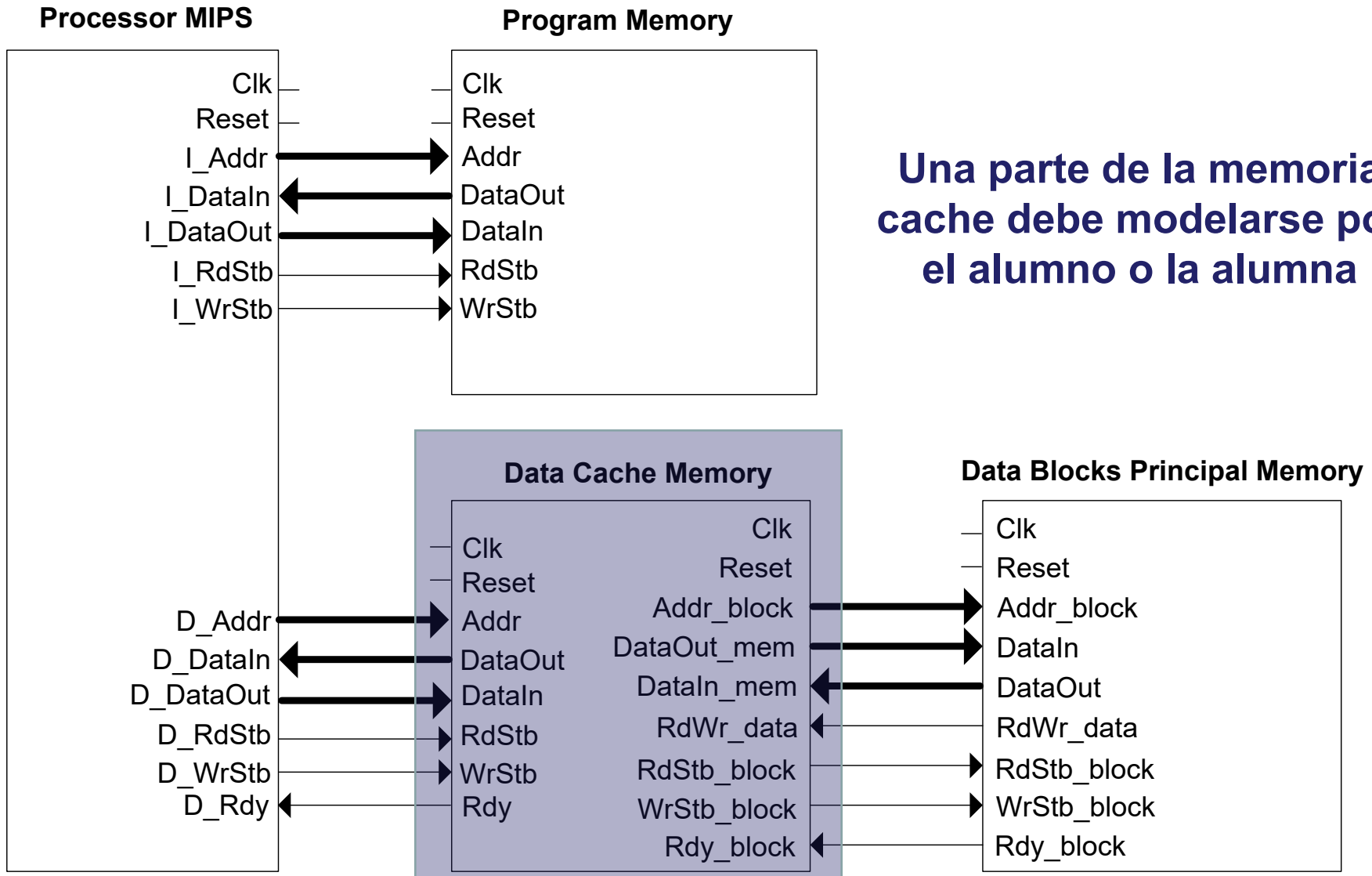
# Diagrama de bloques del TP 3



La descripción de la Memoria Principal no se debe modificar

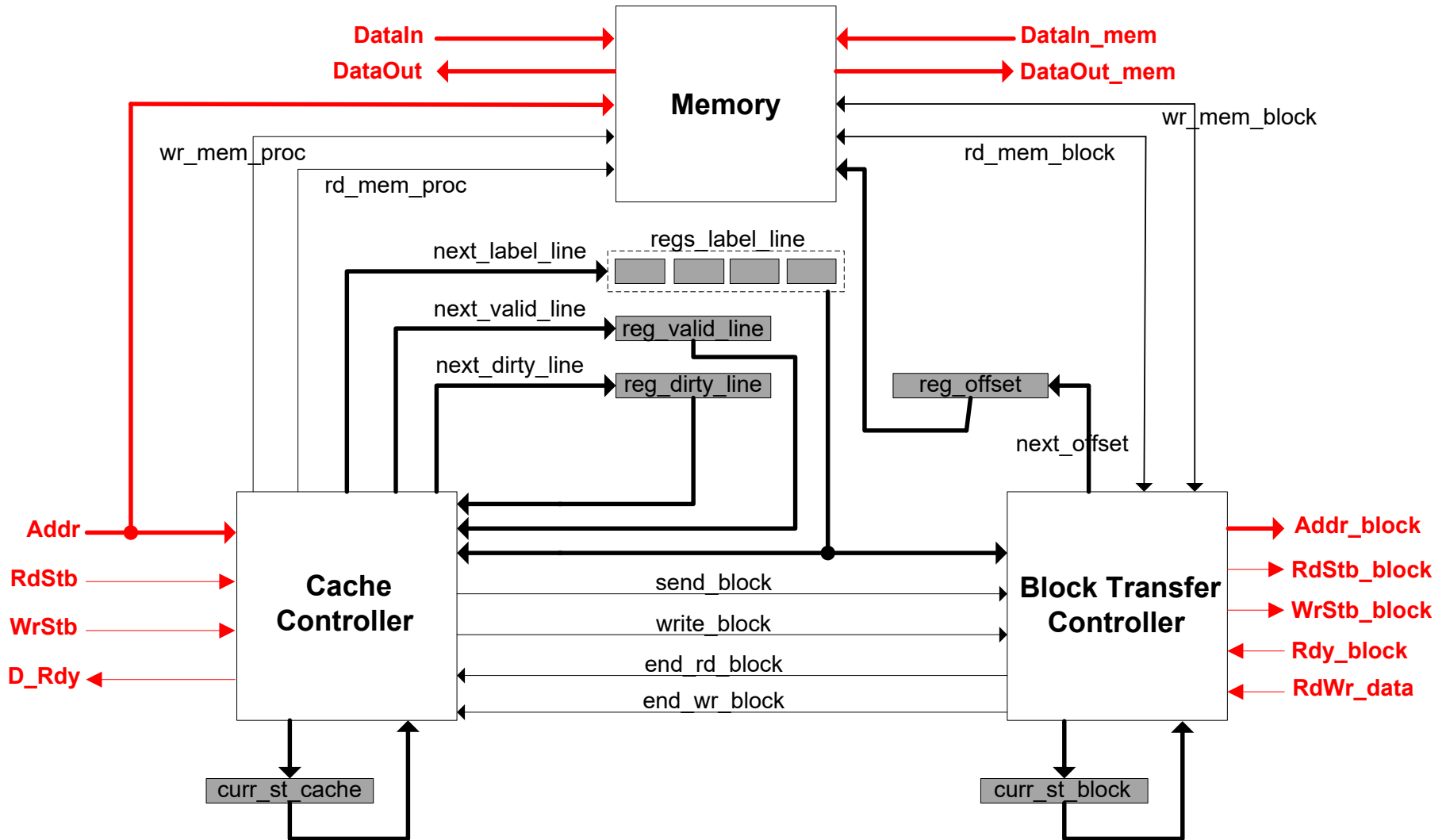
Se carga inicialmente con el archivo de datos *data3*

# Diagrama de bloques del TP 3

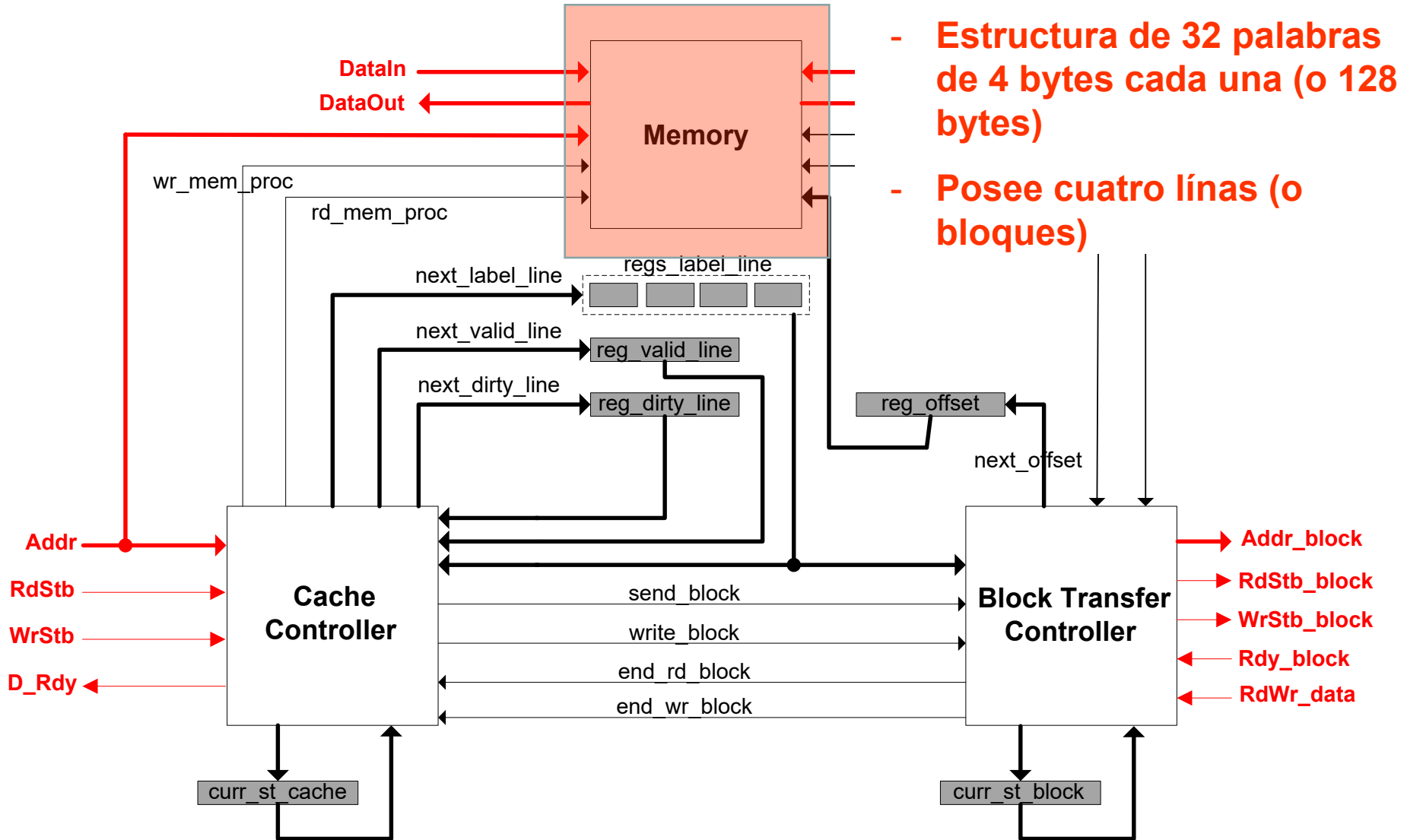


**Una parte de la memoria cache debe modelarse por el alumno o la alumna**

# Data Cache Memory (*MemoryCache*)



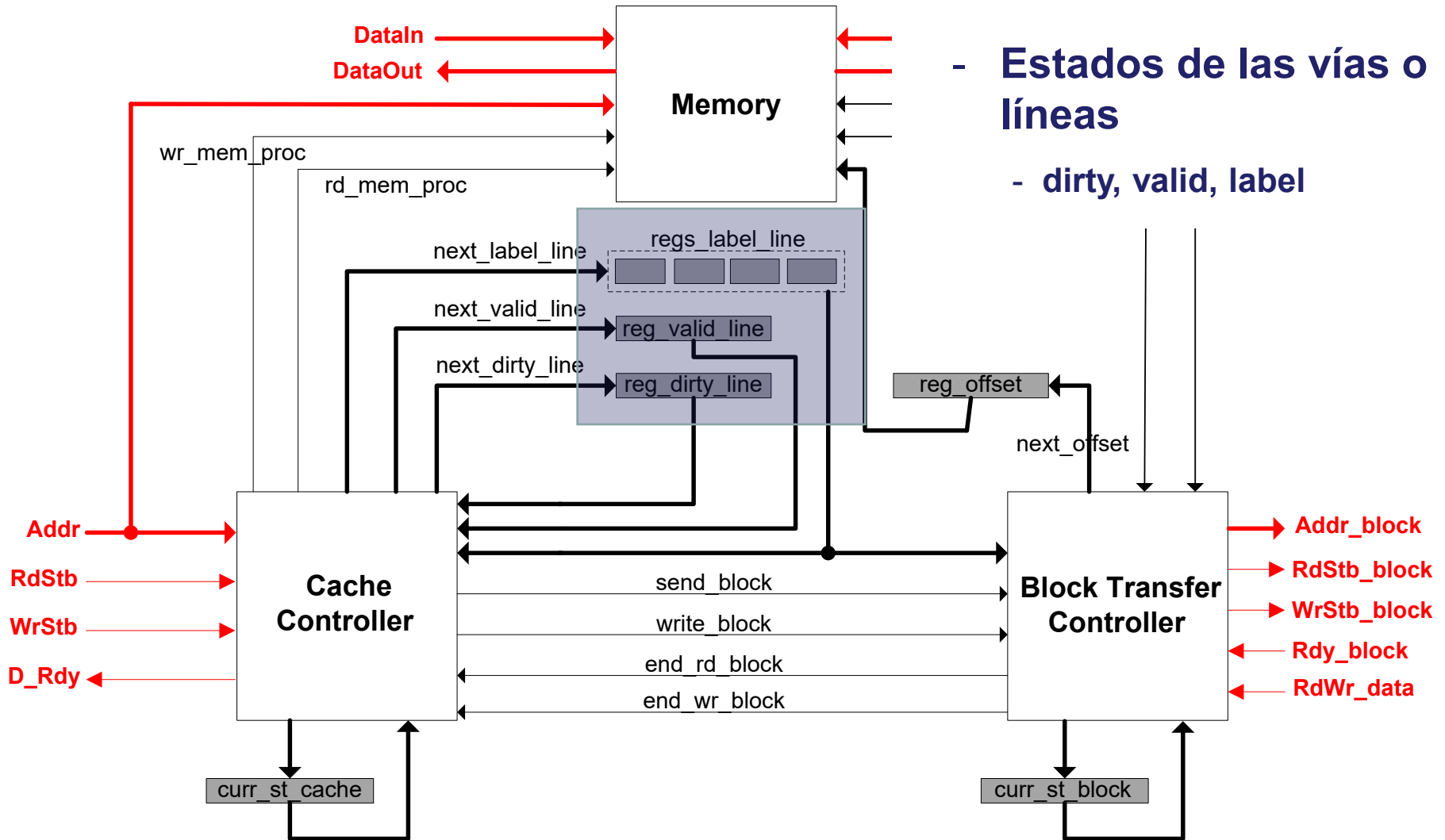
# Data Cache Memory (*MemoryCache*)



- Estructura de 32 palabras de 4 bytes cada una (o 128 bytes)
- Posee cuatro líneas (o bloques)



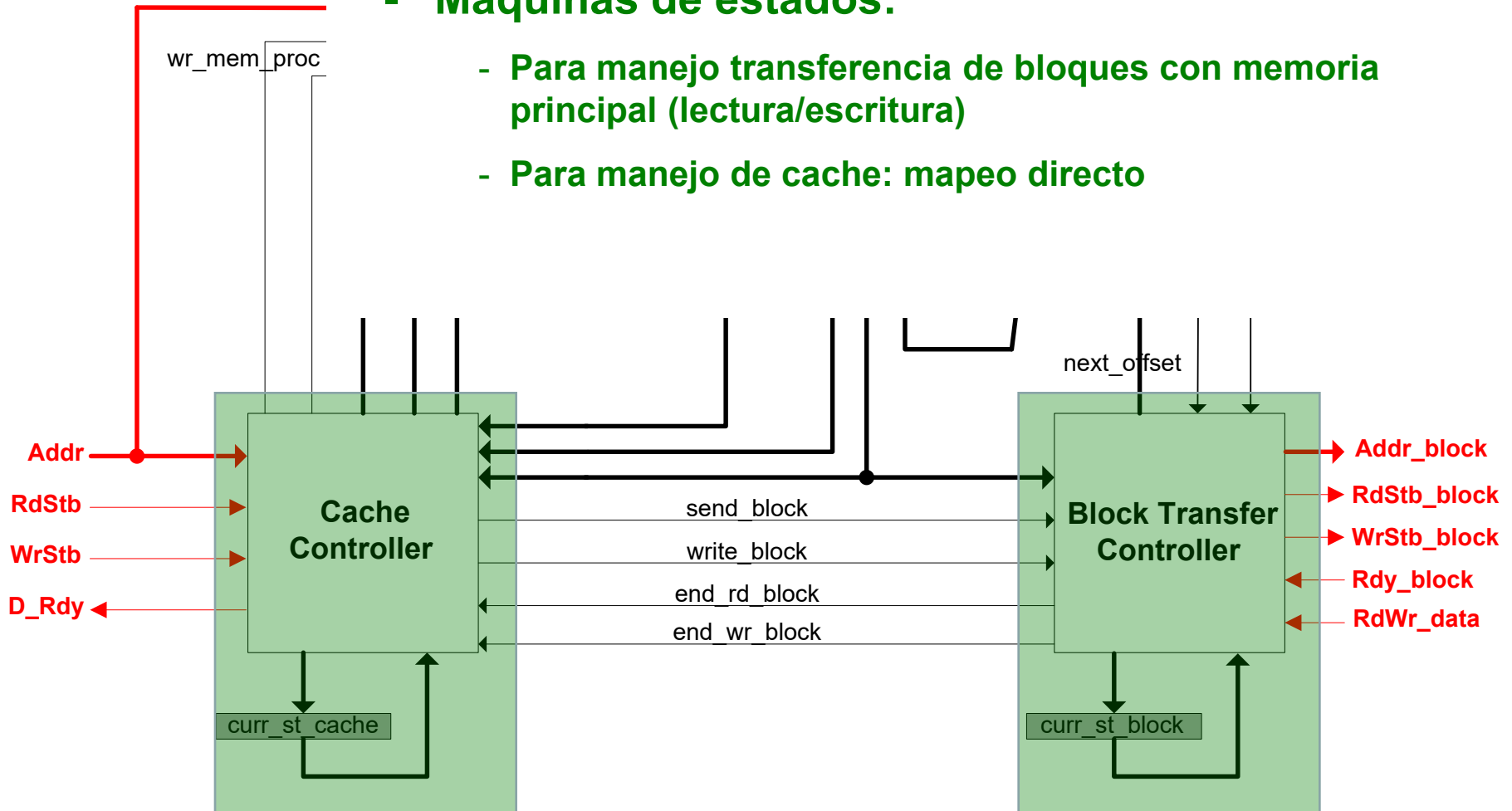
# Data Cache Memory (*MemoryCache*)



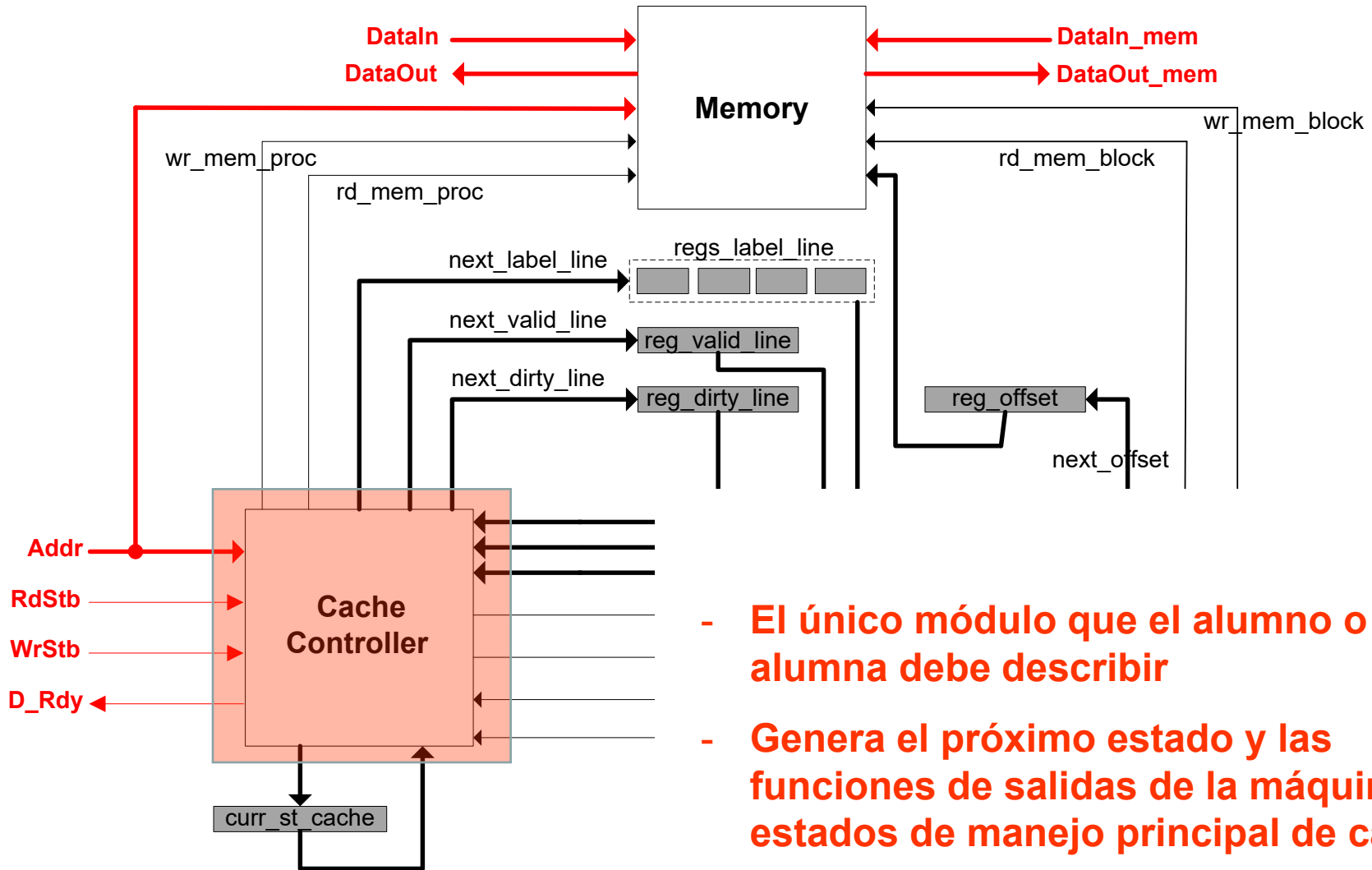
# Data Cache Memory (*MemoryCache*)

## - Máquinas de estados:

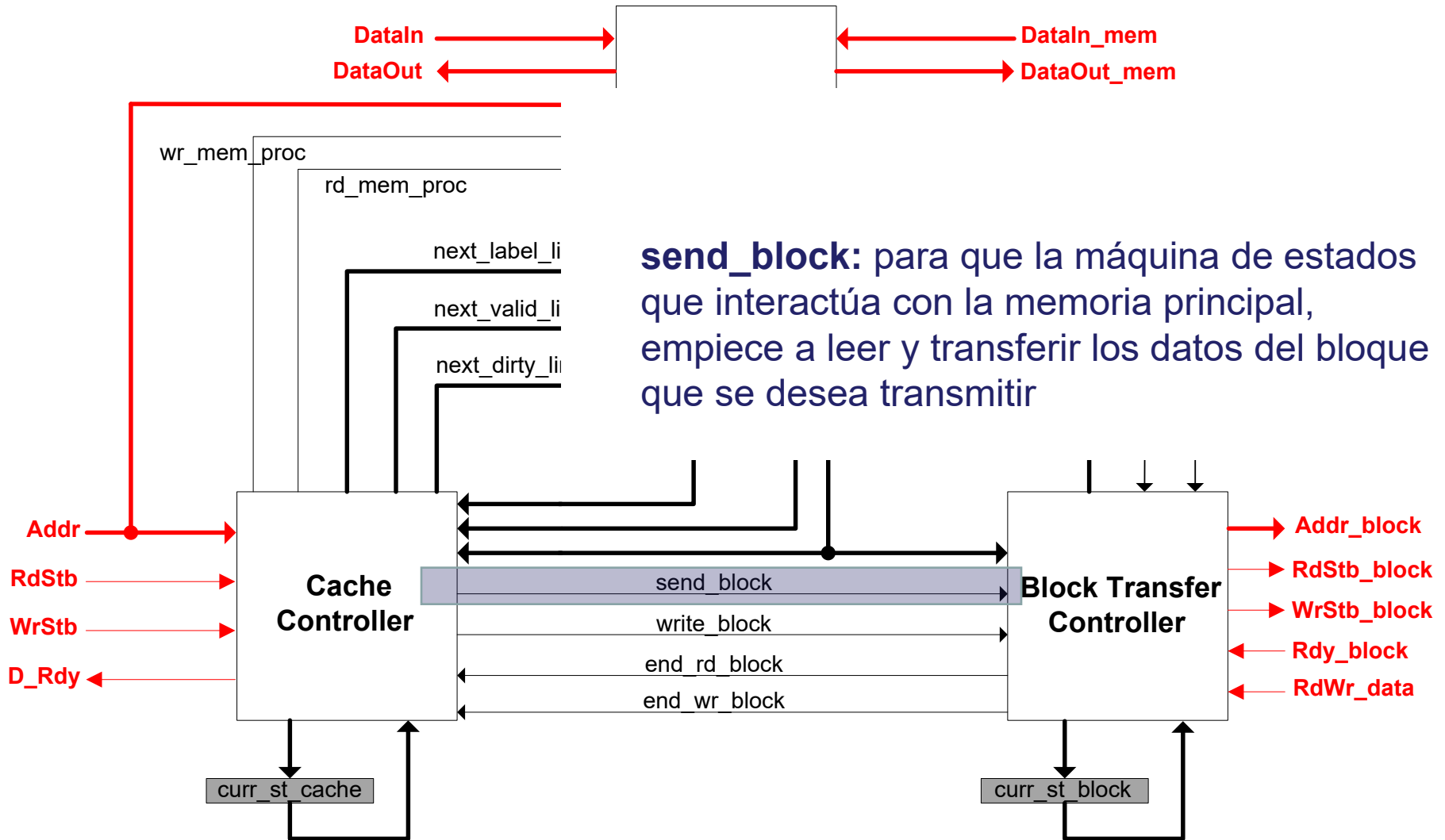
- Para manejo transferencia de bloques con memoria principal (lectura/escritura)
- Para manejo de cache: mapeo directo



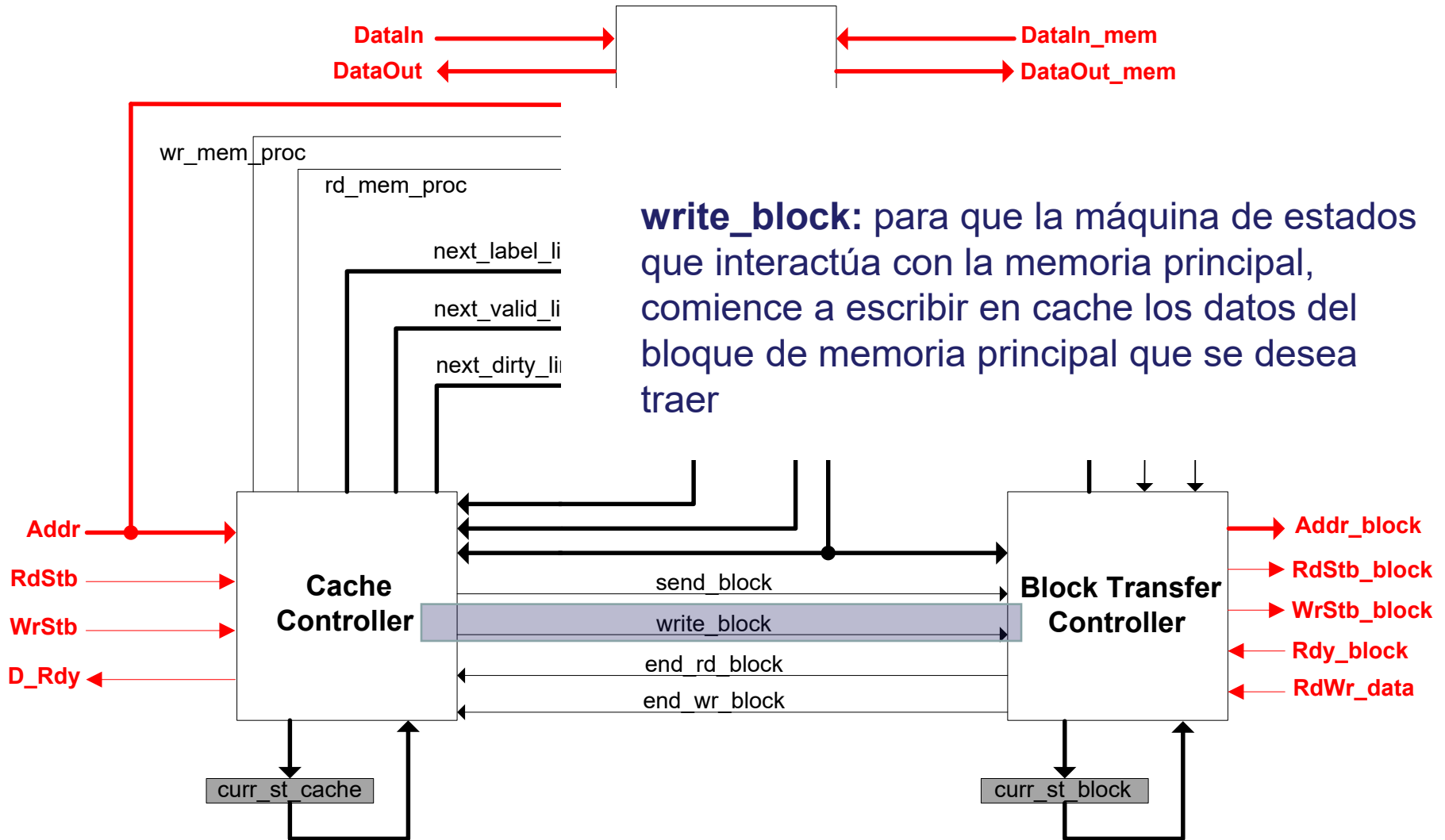
# Data Cache Memory (*MemoryCache*)



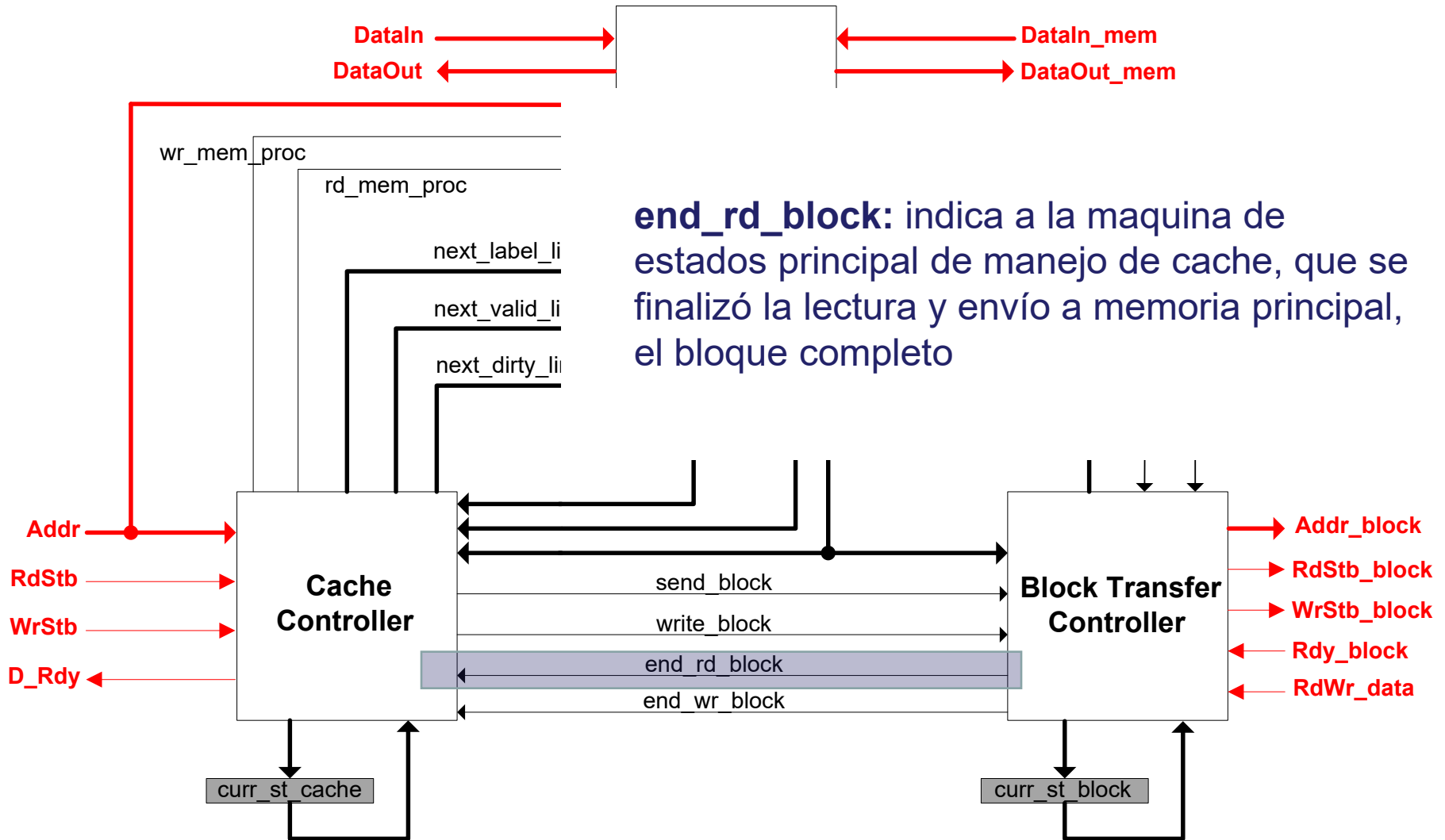
# Data Cache Memory (*MemoryCache*)



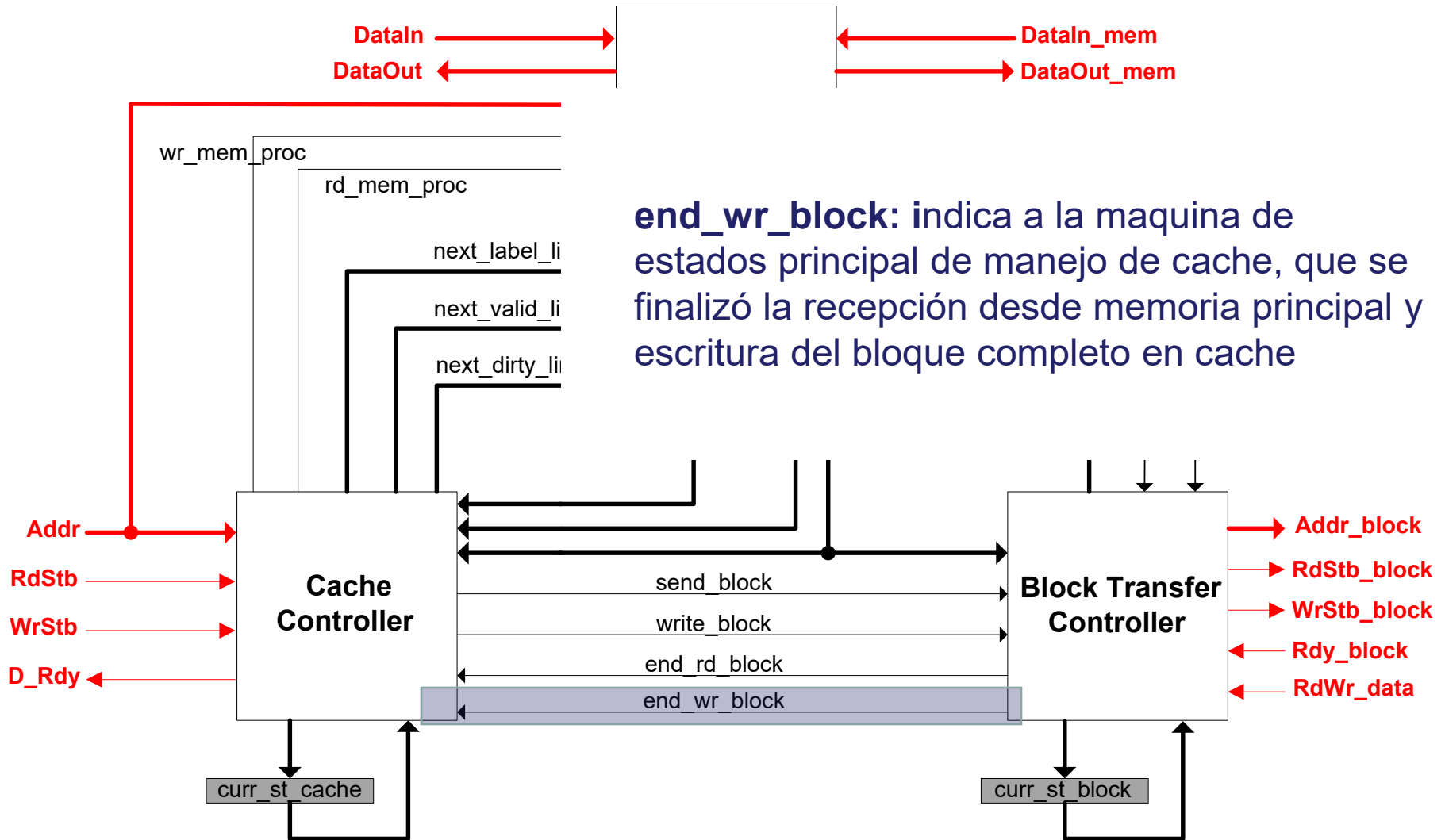
# Data Cache Memory (*MemoryCache*)



# Data Cache Memory (*MemoryCache*)



# Data Cache Memory (*MemoryCache*)



# Acerca de las direcciones en memoria cache

- El tamaño de la estructura interna de la cache es de 4 bloques de 8 palabras (32 palabras o 128 bytes)
  - Para direccionar un bloque se requieren 27 bits
  - En la simulación, los 22 bits más significativos de las direcciones de las memorias deben tener '0's

## Data Cache Memory

Clk	Clk
Reset	Reset
Addr	Addr_block
DataOut	DataOut_mem
DataIn	DataIn_mem
RdStb	RdWr_data
WrStb	RdStb_block
Rdy	WrStb_block
	Rdy_block