



Universidad Tecnológica Nacional
Facultad Regional San Nicolás

Ejemplo de multiprocesamiento

Técnicas Digitales III

Felipe Poblete, Mariano González - 2022

Multiprocesamiento

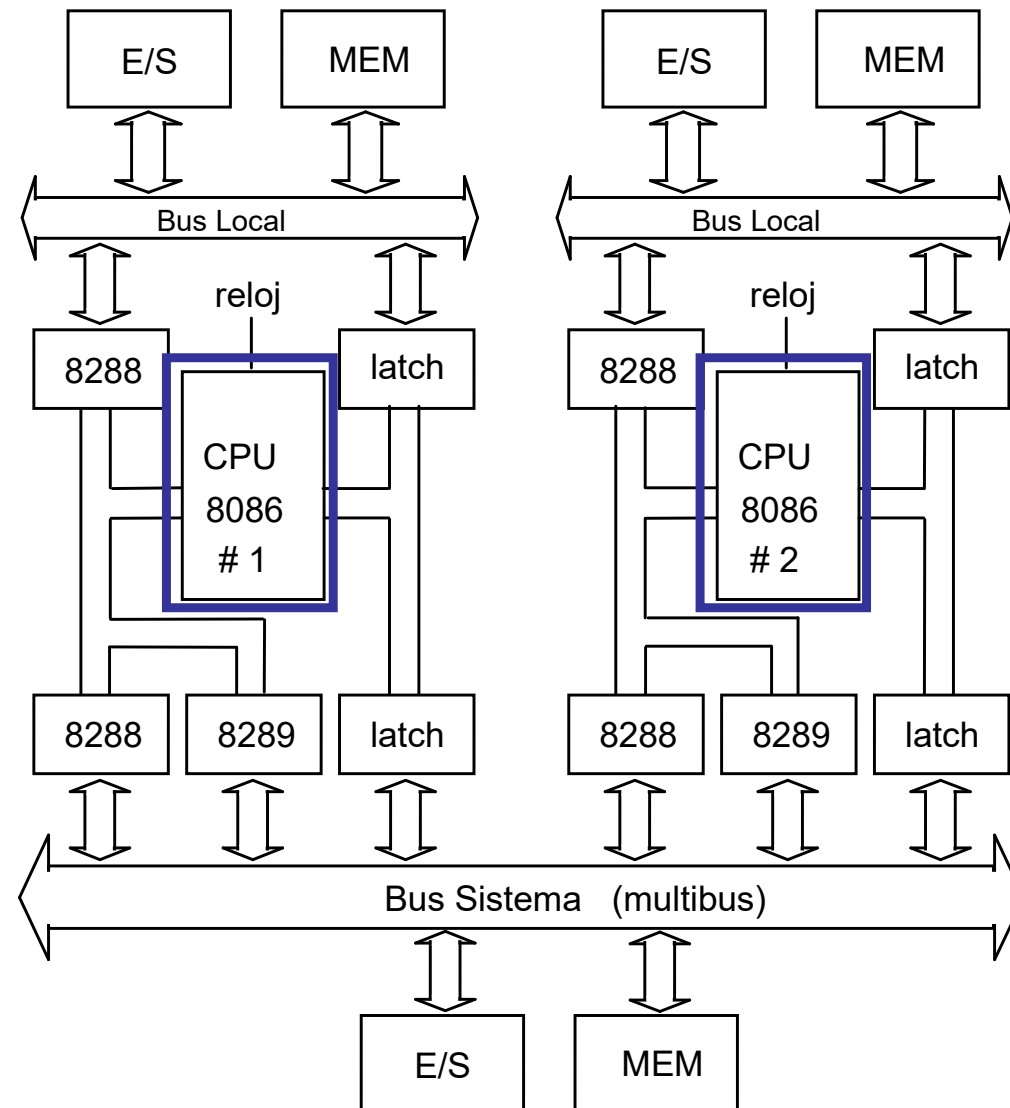
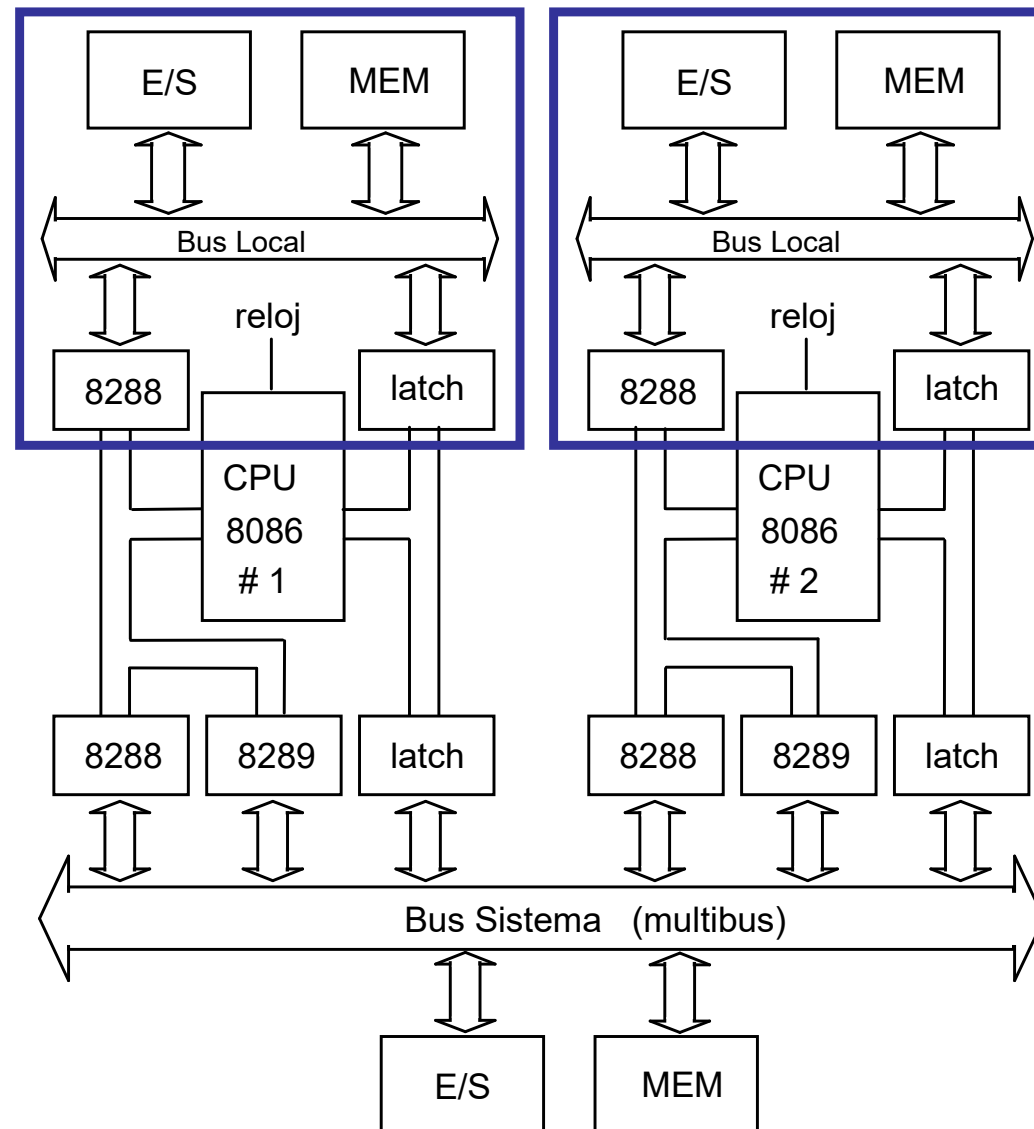


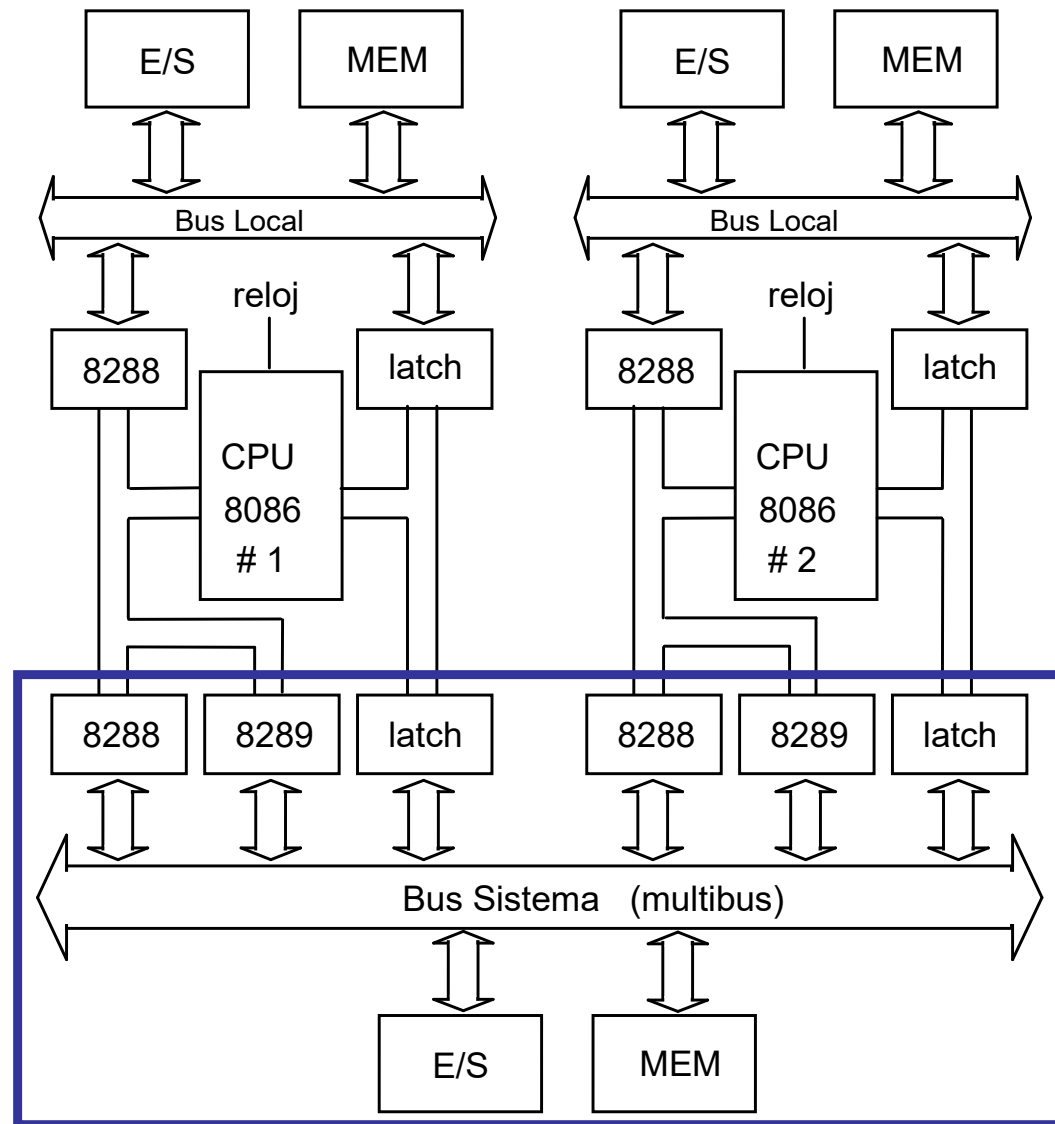
Figura 1 - Sistema con dos procesadores

Multiprocesamiento



Cada procesador posee su bus, memoria y E/S locales.

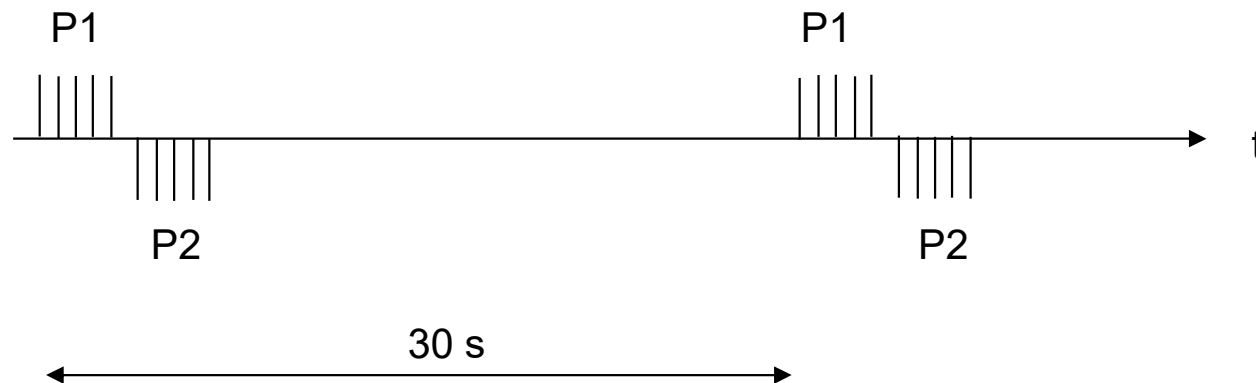
Multiprocesamiento



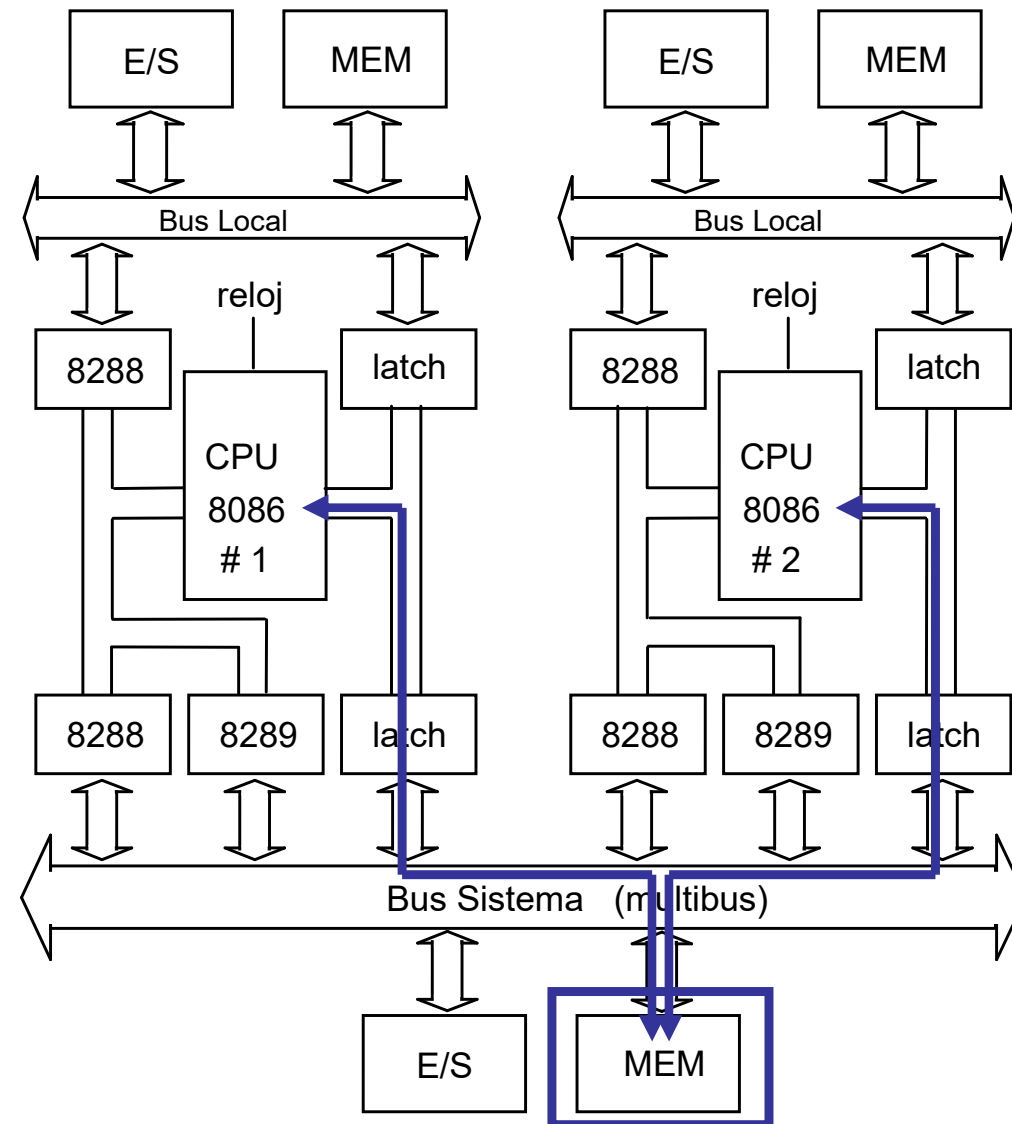
Y ambos comparten un bus, memoria y E/S de sistema.

Multiprocesamiento

Ejemplo: Se trata de un sistema de dos procesadores conectados como indica la figura 1. El primero de ellos (P1) adquiere datos de 5 temperaturas de un proceso cada 30 segundos. Luego de realizar la conversión a °C los debe enviar al otro procesador (P2) para realice los cálculos apropiados de acuerdo a un modelo matemático y evalúe las correcciones necesarias.



Multiprocesamiento



Para comunicar los procesadores se desea utilizar una tabla de 5 valores en la memoria compartida o de sistema

Multiprocesamiento

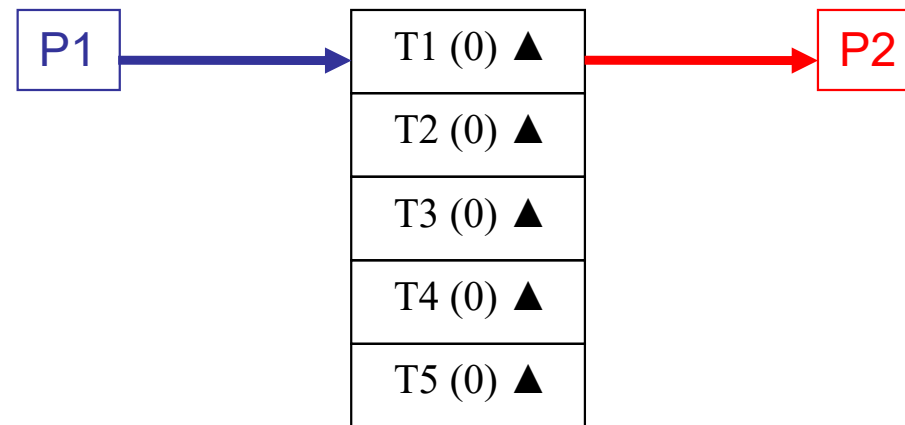


Tabla 1

T1 (0) indica el valor de temperatura 1 adquirido en el muestreo 0. La dirección del primer dato es acordada entre los dos procesadores y las de los restantes siguen en orden consecutivo. El procesador P2 debe poder leer la tabla en forma asincrónica, es decir, cuando lo estime conveniente, sin previo aviso de P1. Los 5 valores deben ser considerados como un “conjunto” de datos, que sólo es válido cuando hayan sido obtenidos en el mismo ciclo de muestreo (▲).

Multiprocesamiento

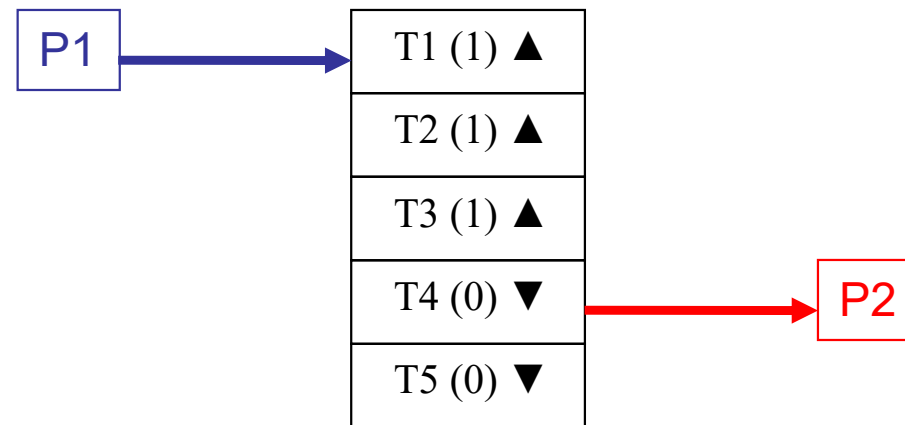


Tabla 2

El inconveniente que se puede presentar es que P2 lea la tabla de datos mientras P1 la está actualizando. En este caso, el conjunto de valores será una mezcla de datos nuevos, adquiridos en el muestreo (1) y viejos (0) (ver Tabla 2). El dato T4 leído por P2 no es válido (▼).

Multiprocesamiento

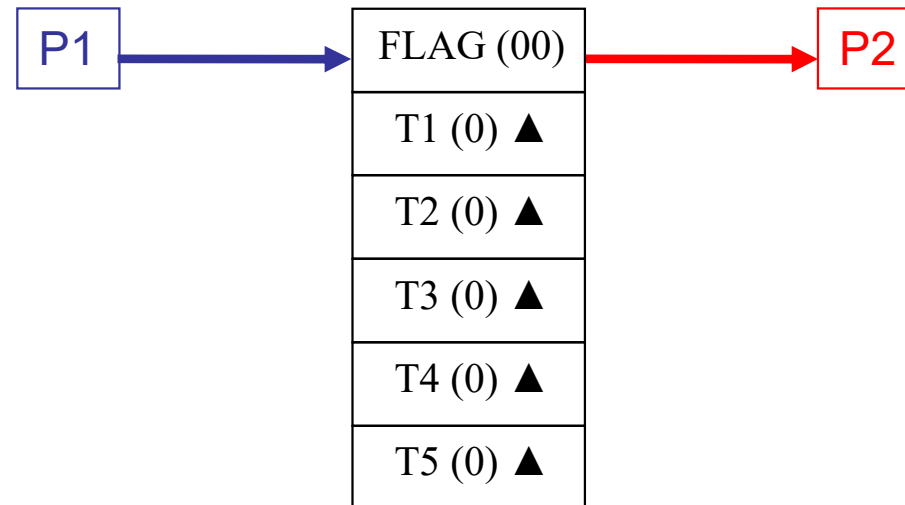


Tabla 3 (habilitada)

Para intentar la solución de este problema aprovecharemos las facilidades de HW y SW de ambos procesadores y agregaremos una posición mas en la tabla, que será una bandera o semáforo para controlar el acceso a la tabla. La misma en 00 indica que el conjunto de datos es coherente y corresponde al mismo muestreo. En este estado puede ser leído por P2. Con criterio mas amplio, la tabla está “habilitada”.

Multiprocesamiento

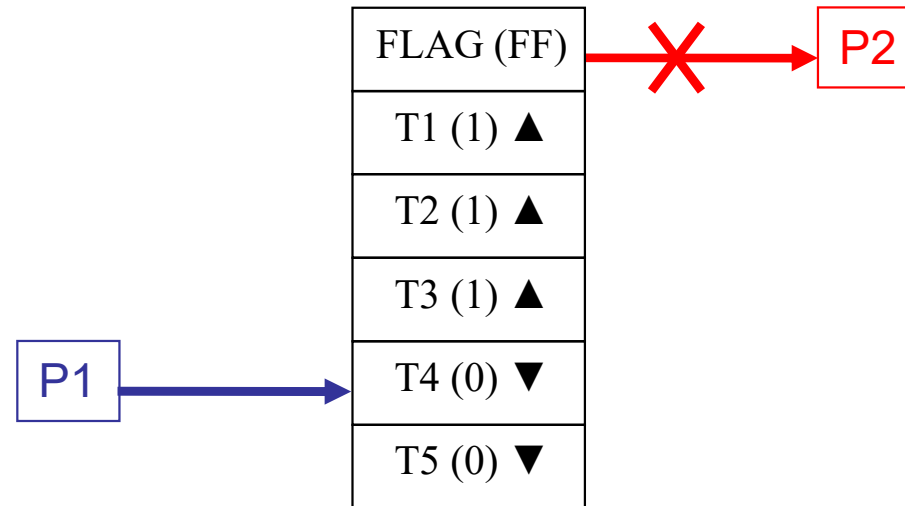


Tabla 4 (bloqueada)

La bandera en FF indica que la tabla está siendo actualizada por P1 o leída por P2. En este estado, podemos decir que la tabla está “bloqueada” por el procesador que la está accediendo.

Multiprocesamiento

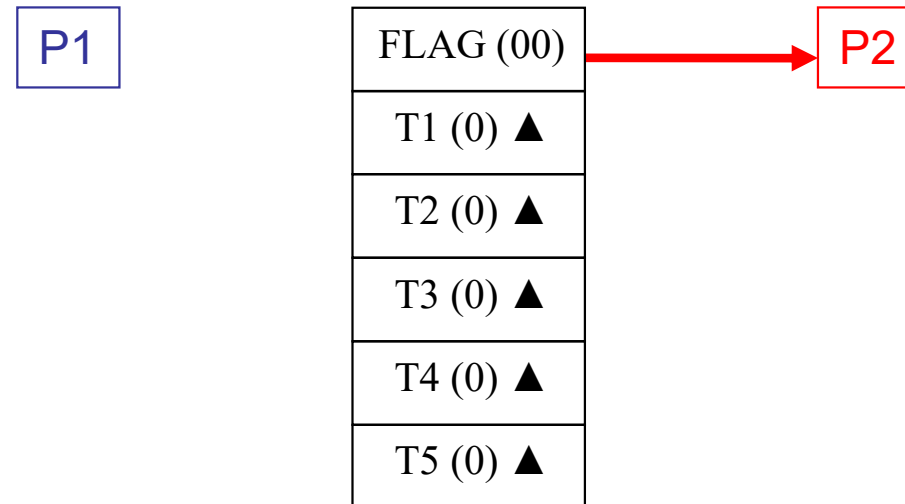
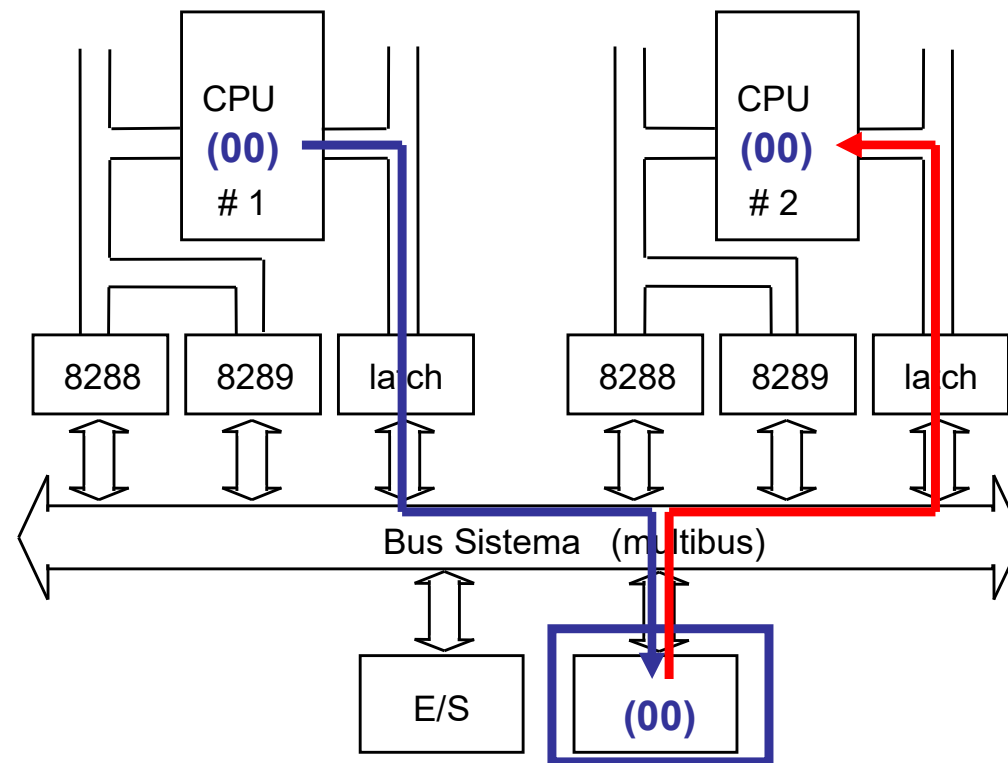


Tabla 3 (habilitada)

Partimos de la situación en que los datos son coherentes y FLAG=00. Cuando P2 decide leer, verifica primero si FLAG=00. Si es así, pone FLAG=FF y luego lee los 5 datos. Si en el primer intento FLAG=FF, espera e intenta nuevamente. Luego de leer todos los datos, P2 pone FLAG=00. Cuando P1 necesita actualizar los datos, verifica primero si FLAG=00. Si es así, pone FLAG=FF, escribe los 5 datos y pone FLAG=00.

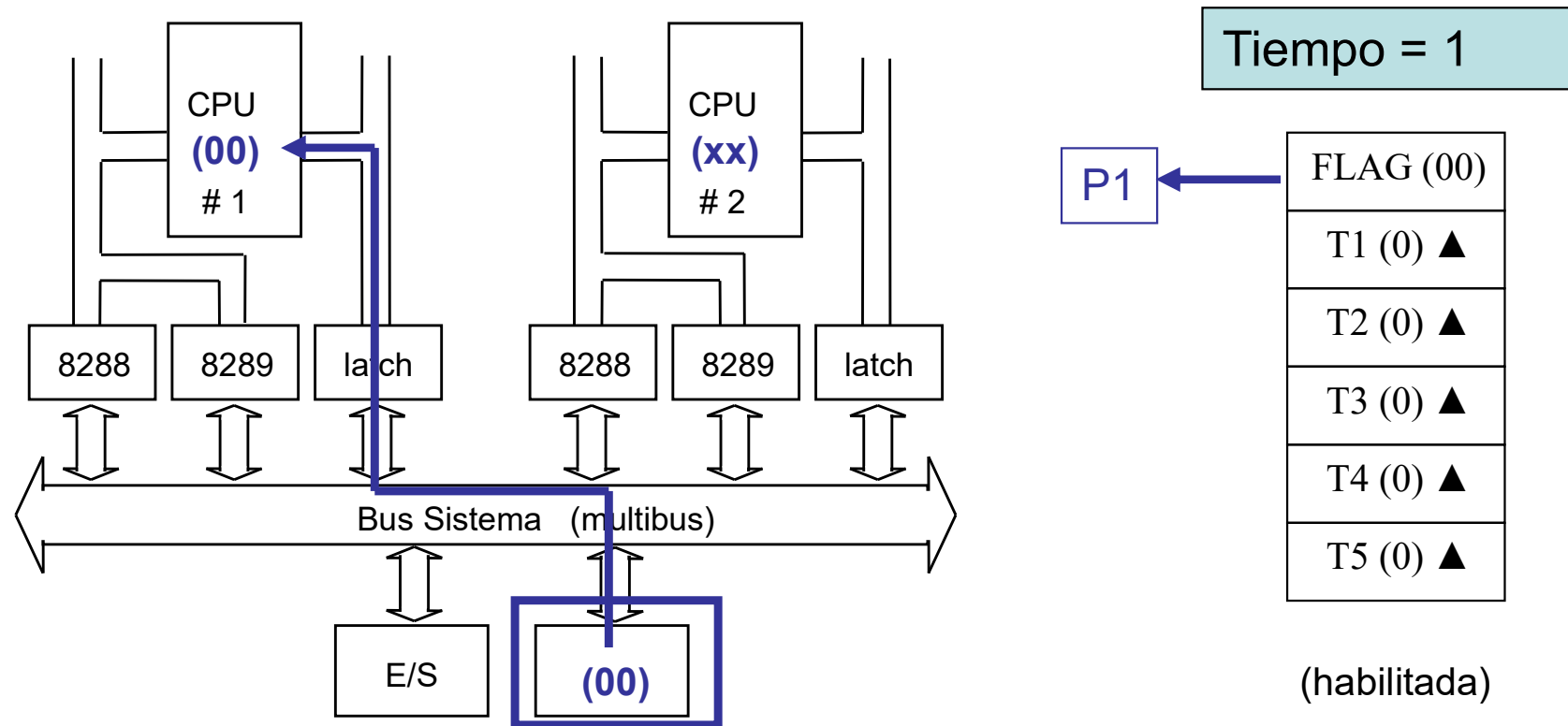
Multiprocesamiento



Parecería que con este simple artilugio de SW es suficiente para solucionar nuestro problema, sin embargo no es así:

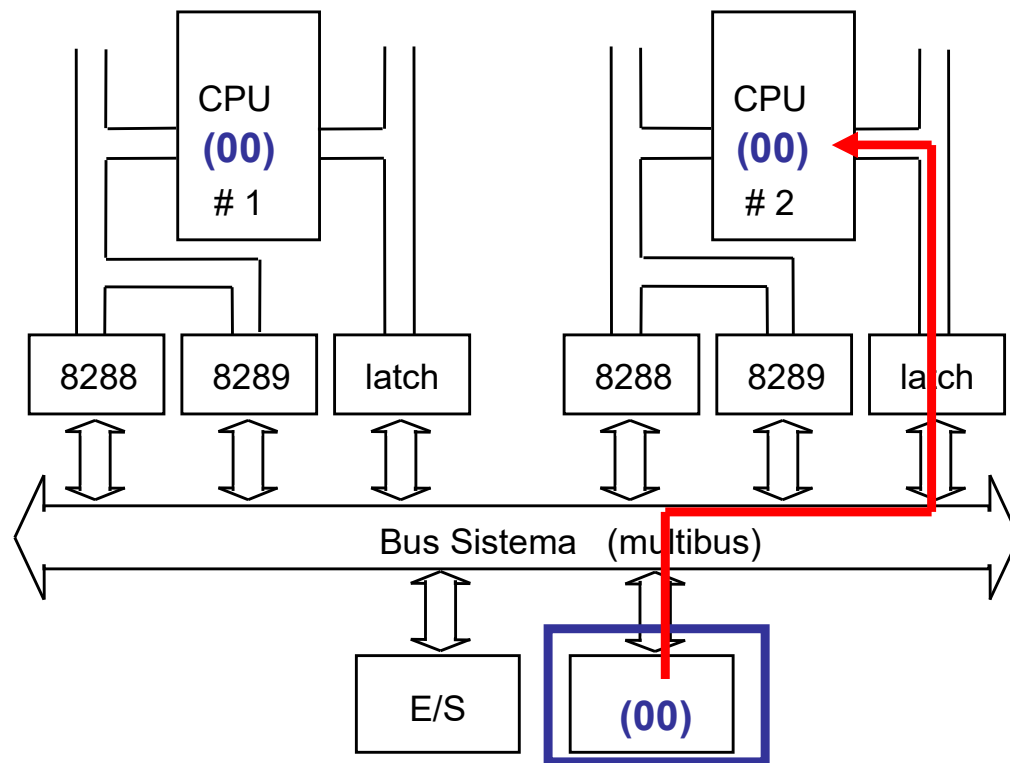
P1 y P2 leen la memoria de sistema y la llevan a un registro interno donde determinarán su valor (00 o FF). Obsérvese que P1 y P2 no pueden utilizar el bus al mismo tiempo y además, la decisión de leer o no en función del valor de FLAG no se puede tomar en un solo ciclo de reloj.

Multiprocesamiento



Haremos la secuencia paso a paso, avanzando con cada ciclo de reloj. El estado inicial es la tabla habilitada. Ambos procesadores copian FLAG a un registro interno, pero no lo pueden hacer a la vez. Supongamos que P1 lo hace primero, leyendo (00).

Multiprocesamiento



Tiempo = 2

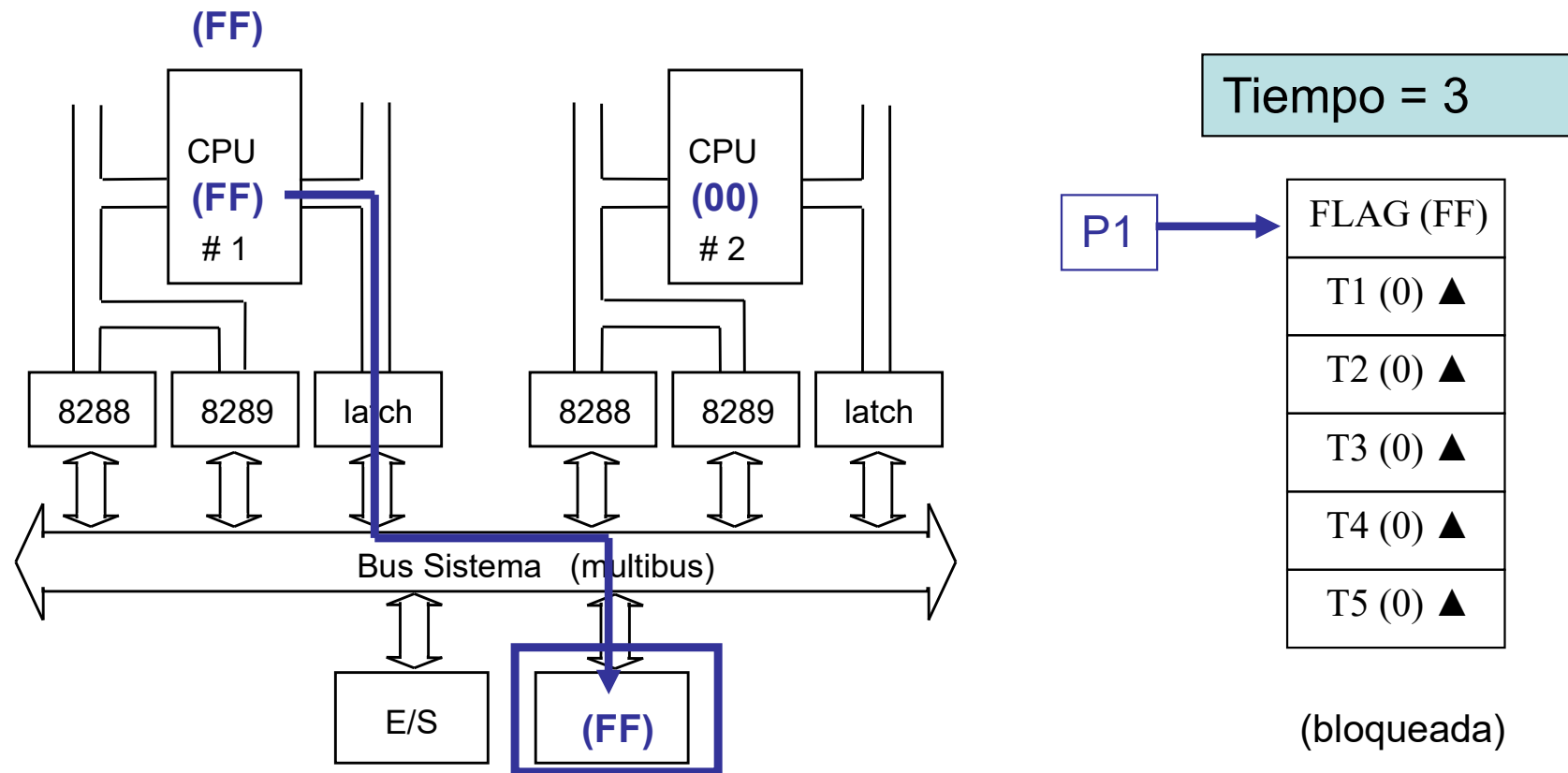
FLAG (00)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

P2

(habilitada)

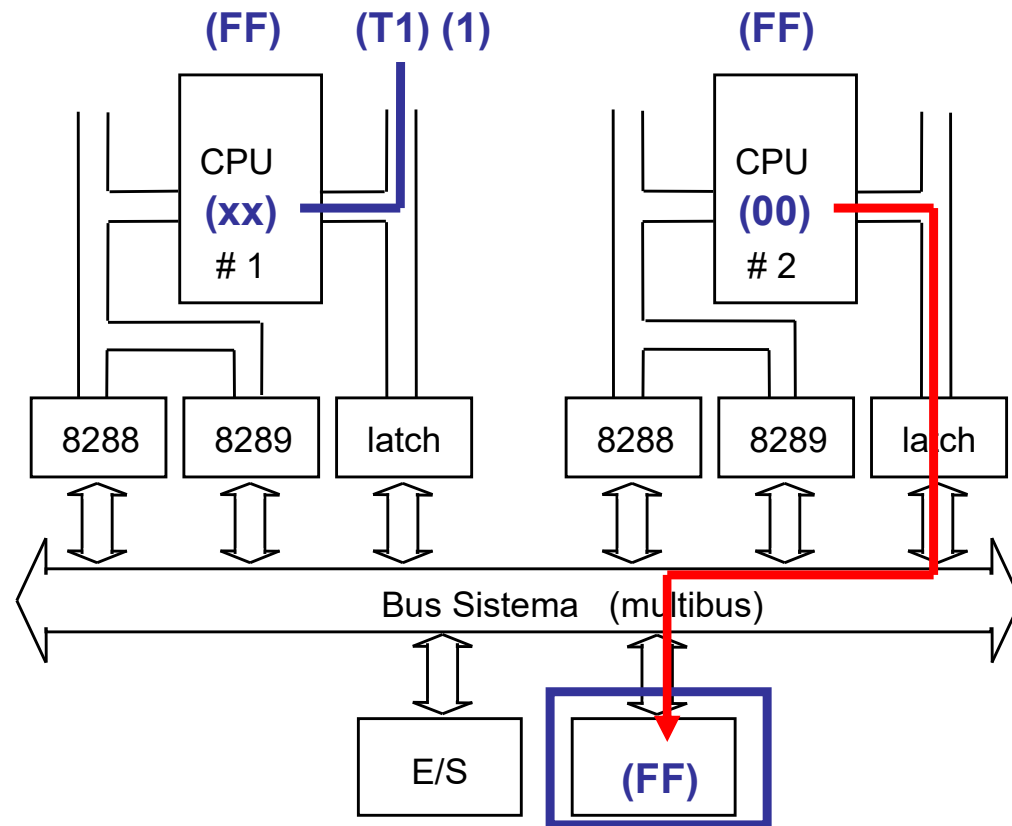
Ahora lee P2, el mismo valor (00), ya que al ser el siguiente ciclo de reloj P1 no pudo aún bloquear la tabla.

Multiprocesamiento



P1 bloquea la tabla escribiendo FF, registra ese estado en su memoria local y se dispone a acceder a la tabla para actualizar los valores de T1 a T5.

Multiprocesamiento



Tiempo = 4

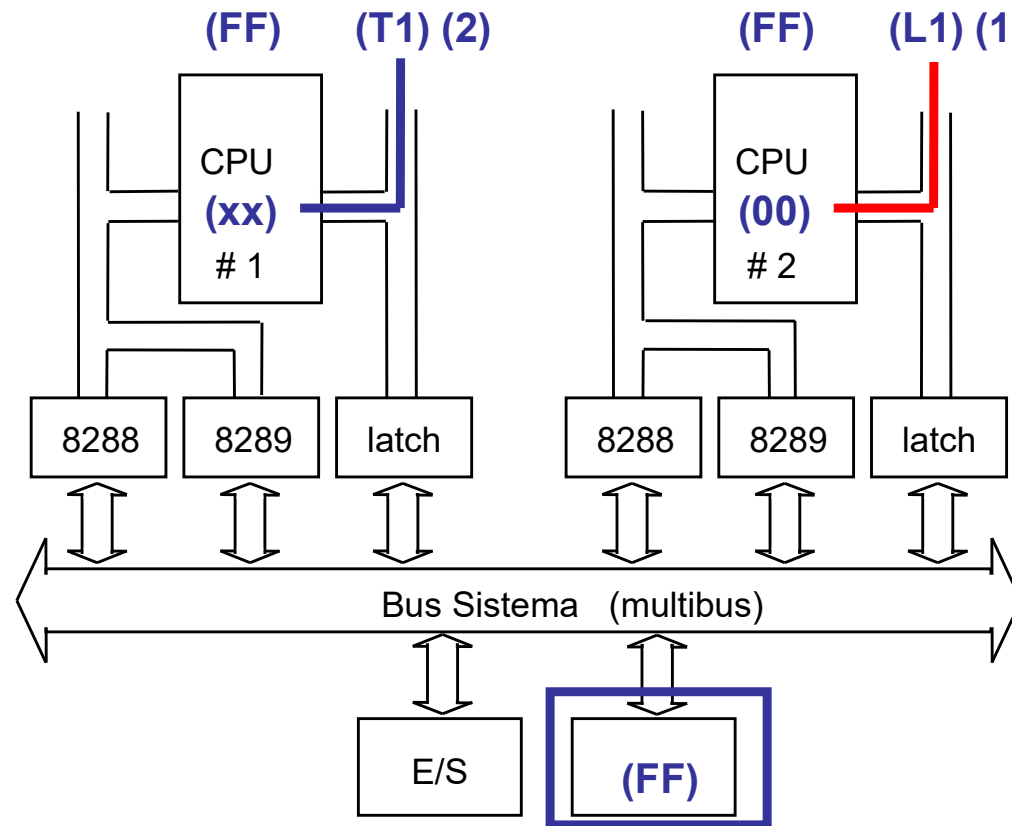
FLAG (00)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

P2

(bloqueada)

Ahora lee P2, el mismo valor (00), ya que al ser el siguiente ciclo de reloj P1 no pudo aún bloquear la tabla. Al mismo tiempo, P1 realiza el cálculo de T1 y se dispone a escribirlo en la tabla. Supusimos que este cálculo demora 3 unidades de tiempo.

Multiprocesamiento



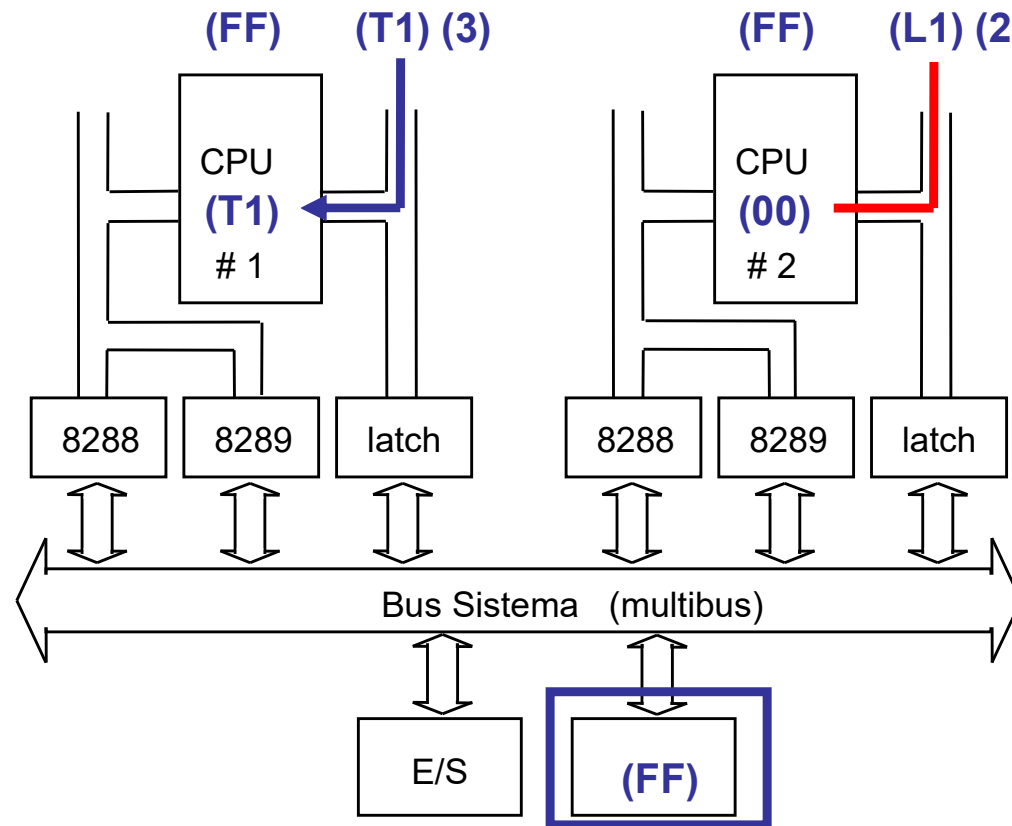
Tiempo = 5

FLAG (FF)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

(bloqueada)

P1 está en su segundo ciclo de cálculo y P2 se prepara para leer los valores de la tabla. A esta tarea le asignamos tres unidades de tiempo.

Multiprocesamiento



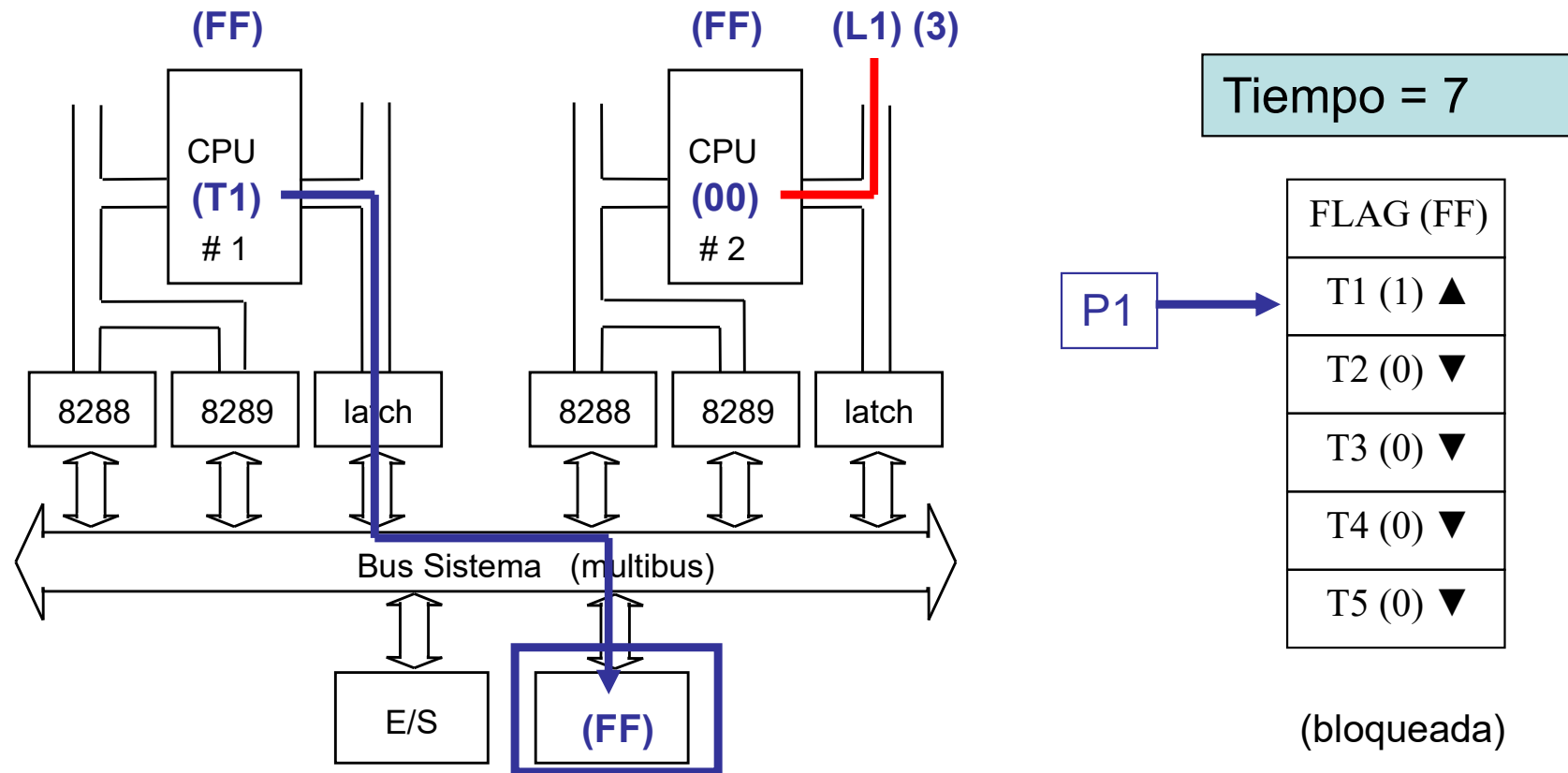
Tiempo = 6

FLAG (FF)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

(bloqueada)

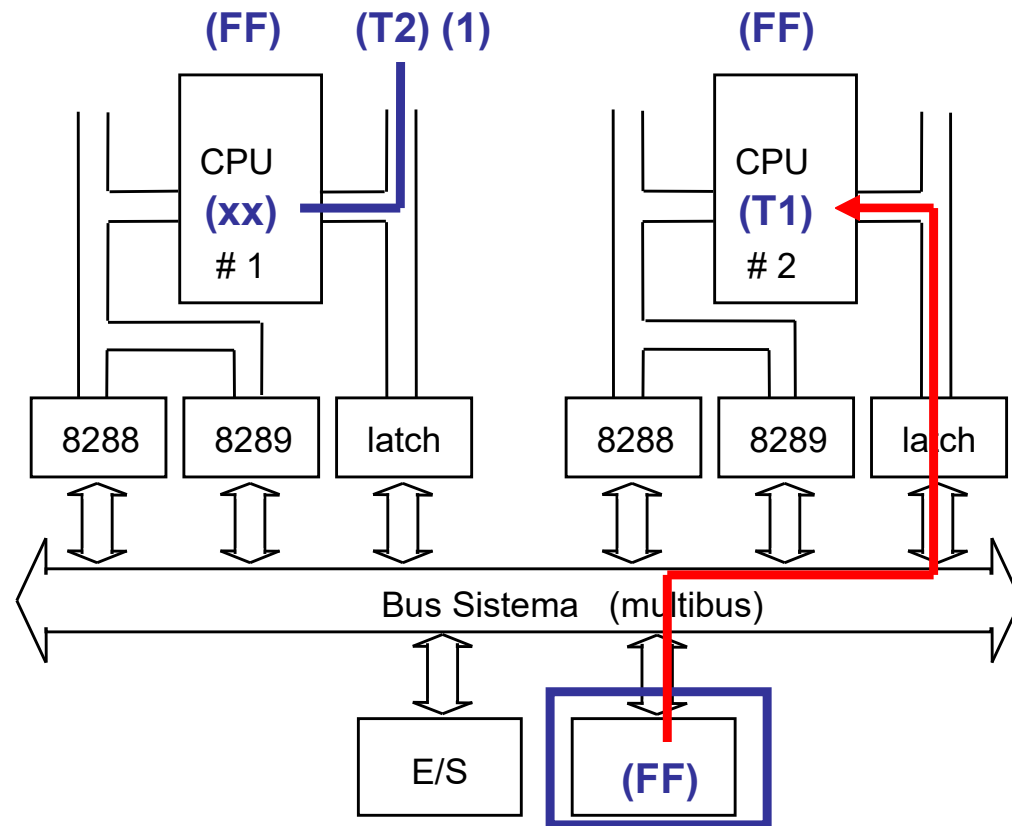
P1 está en su último ciclo de cálculo y P2 en su segundo paso de preparación para leer los valores de la tabla. Los símbolos ▲ indican que estos valores son coherentes, ya que corresponden al mismo muestreo.

Multiprocesamiento



P1 escribe el valor T1 en la tabla (recordemos que para él está accesible porque fue quien la bloqueó), mientras P2 está en su último paso de preparación para leer los valores de la tabla. Los símbolos ▼ indican que estos valores no son coherentes, ya que corresponden a diferente muestreo.

Multiprocesamiento



Tiempo = 8

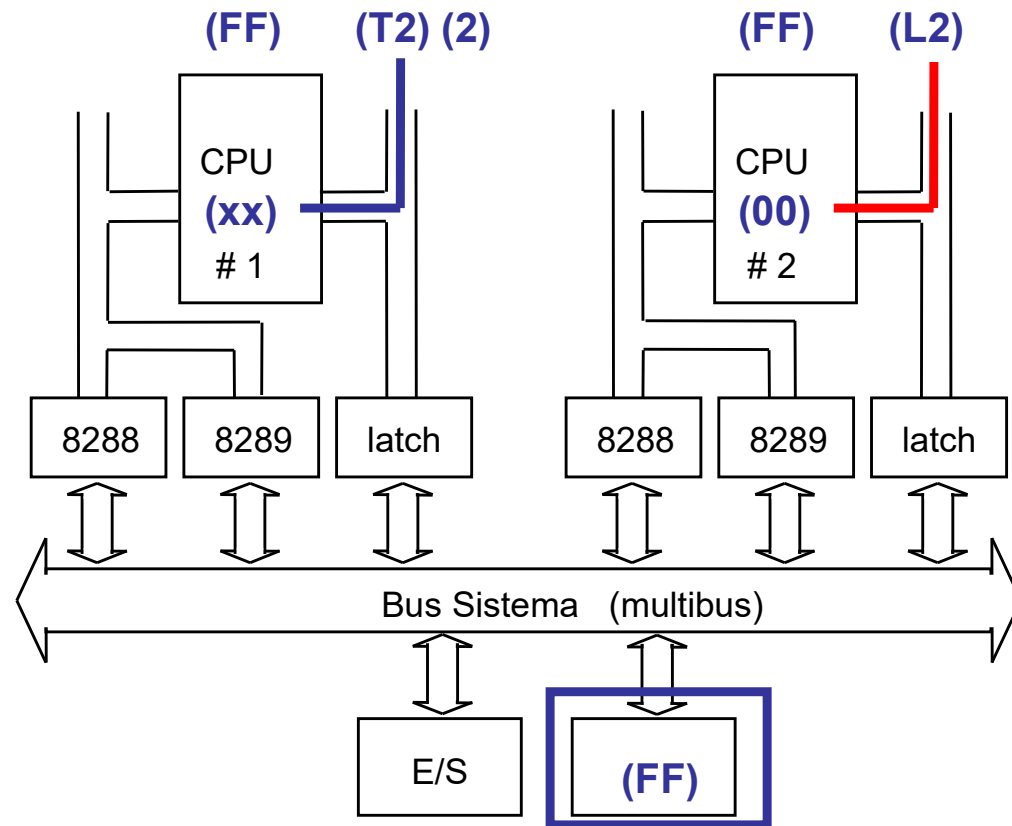
FLAG (00)
T1 (1) ▲
T2 (0) ▼
T3 (0) ▼
T4 (0) ▼
T5 (0) ▼

P2

(bloqueada)

P1 realiza el cálculo de T2 y se dispone a escribirlo en la tabla. Supusimos que este cálculo demora 3 unidades de tiempo. Al mismo tiempo, P2 lee el valor de T1 desde la tabla, que para él está habilitada, ya que él la bloqueó.

Multiprocesamiento



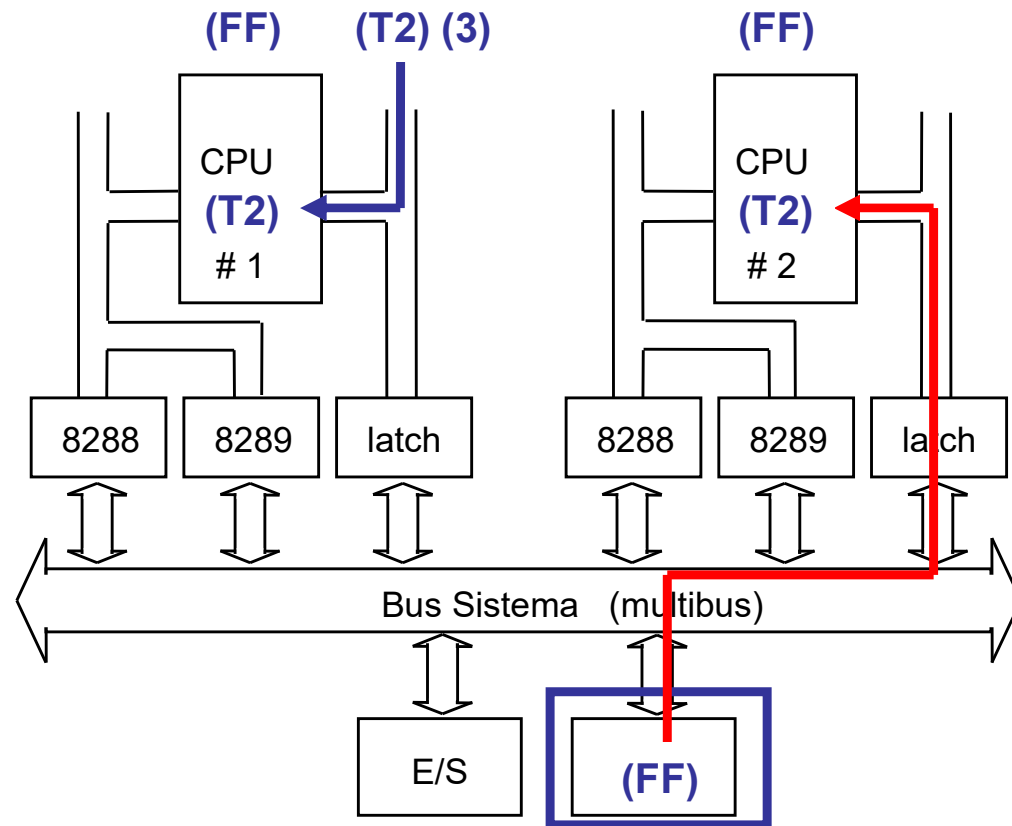
Tiempo = 9

FLAG (FF)
T1 (1) ▲
T2 (0) ▼
T3 (0) ▼
T4 (0) ▼
T5 (0) ▲

(bloqueada)

P1 está en su segundo ciclo de cálculo y P2 se prepara para leer el próximo valor de la tabla. A esta tarea le asignamos una unidad de tiempo.

Multiprocesamiento



Tiempo = 10

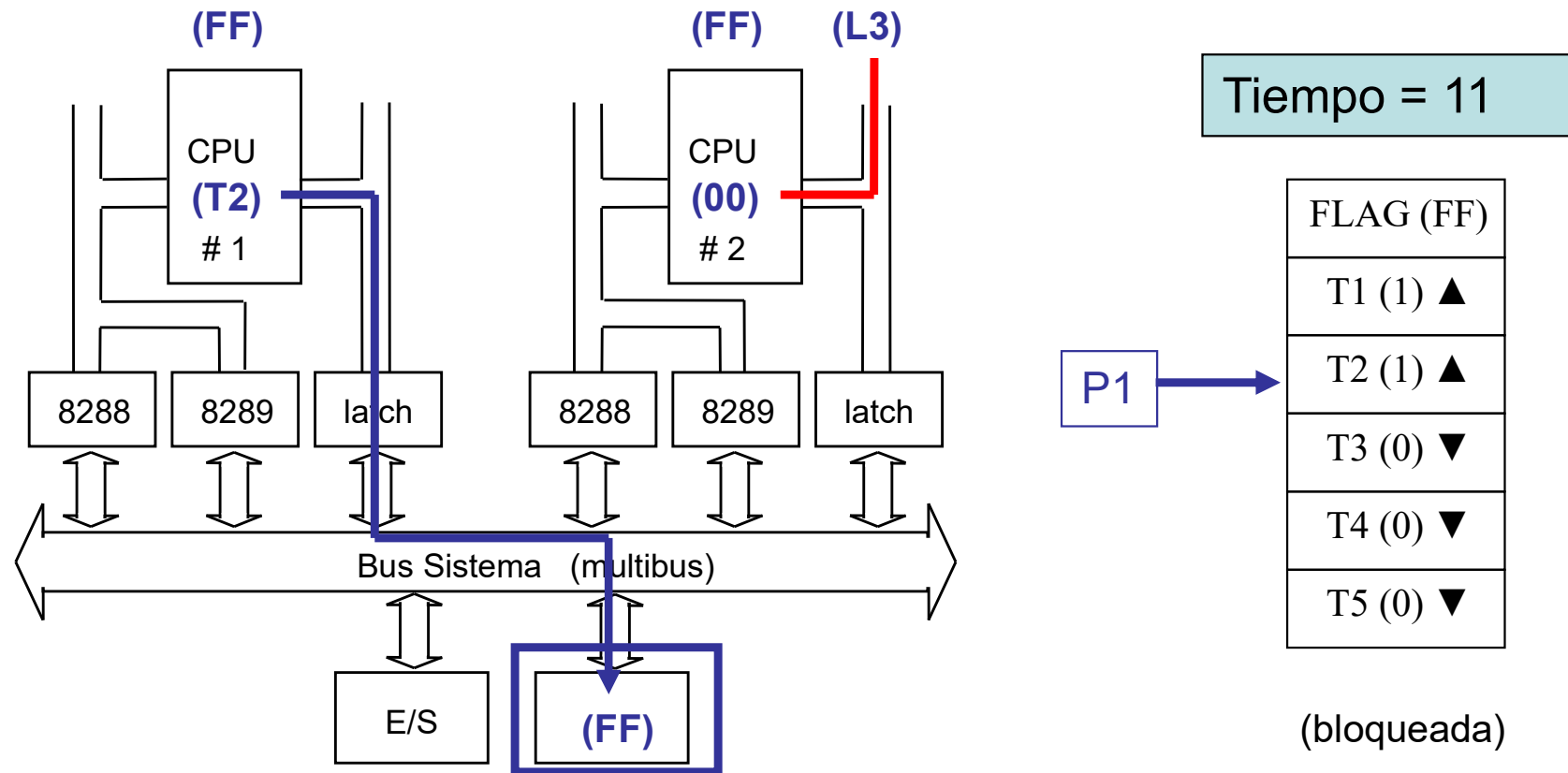
FLAG (00)
T1 (1) ▲
T2 (0) ▼
T3 (0) ▼
T4 (0) ▼
T5 (0) ▼

P2

(bloqueada)

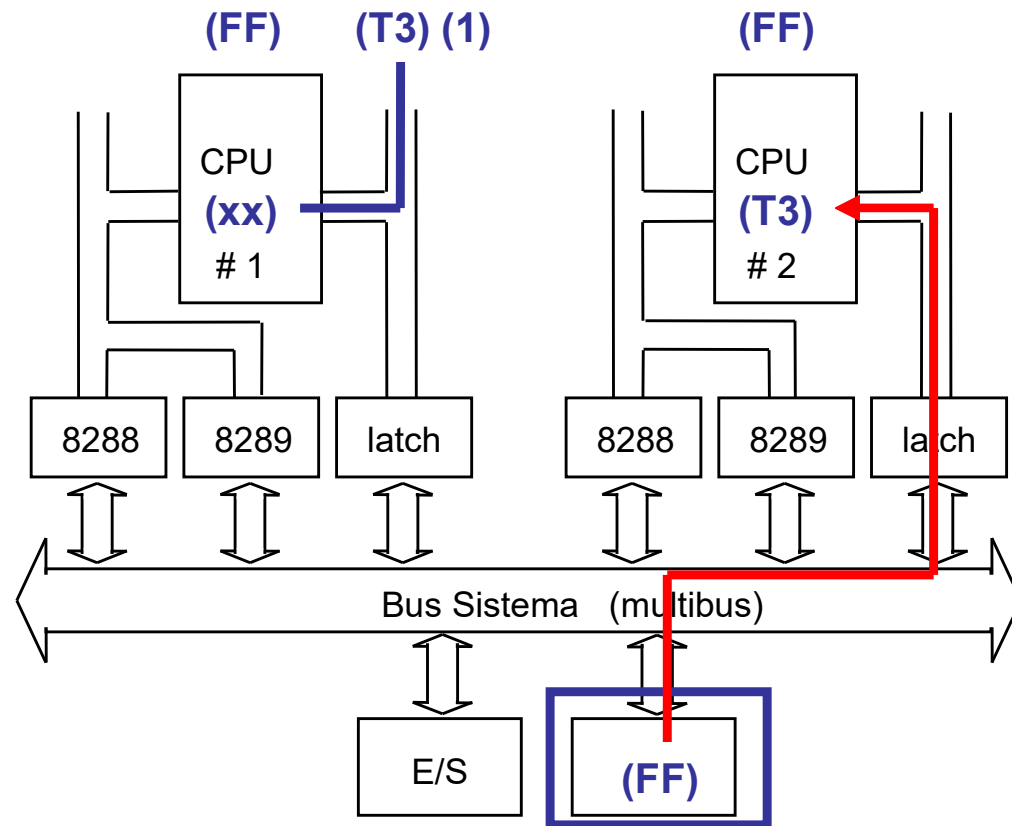
P1 está en su último ciclo de cálculo. Al mismo tiempo, P2 lee el valor de T2 desde la tabla. Observe que este dato no corresponde al mismo muestreo que el anterior.

Multiprocesamiento



P1 escribe el valor T2 en la tabla, mientras P2 está en su último paso de preparación para leer los valores de la tabla.

Multiprocesamiento



Tiempo = 12

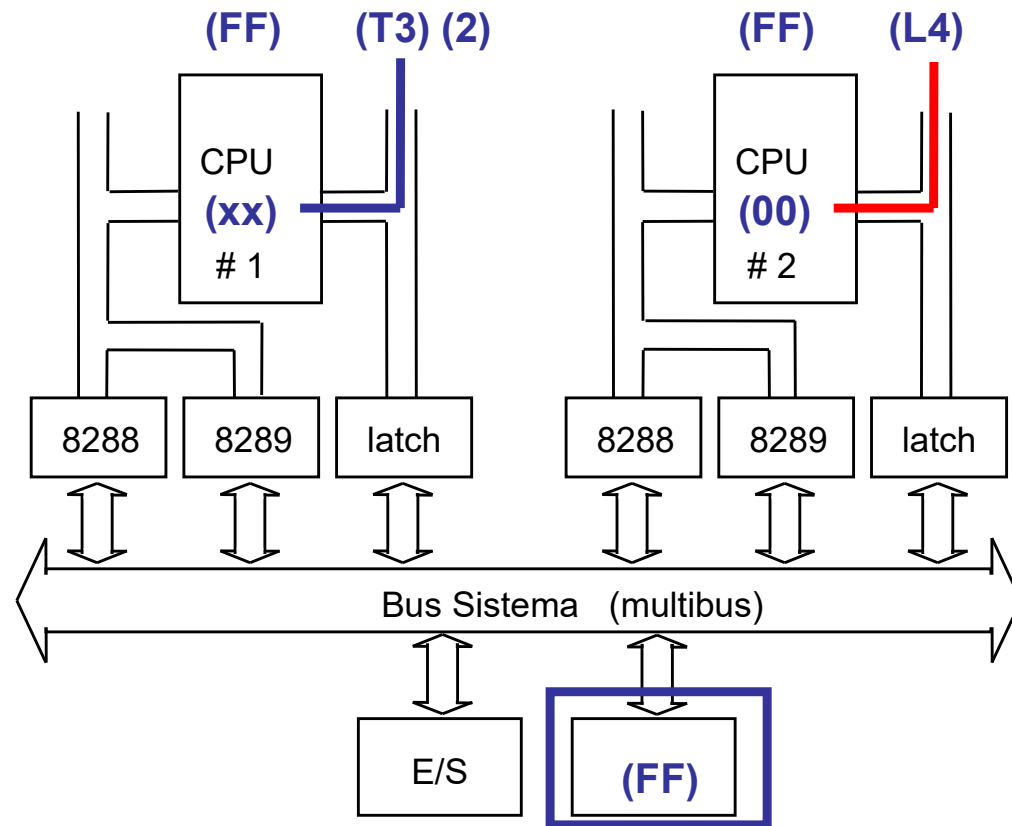
FLAG (00)
T1 (1) ▲
T2 (1) ▲
T3 (0) ▼
T4 (0) ▼
T5 (0) ▼

P2

(bloqueada)

P1 realiza el cálculo de T3 y se dispone a escribirlo en la tabla. Supusimos que este cálculo demora 3 unidades de tiempo. Al mismo tiempo, P2 lee el valor de T3 desde la tabla. Observe que este dato no corresponde al mismo muestreo que el anterior.

Multiprocesamiento



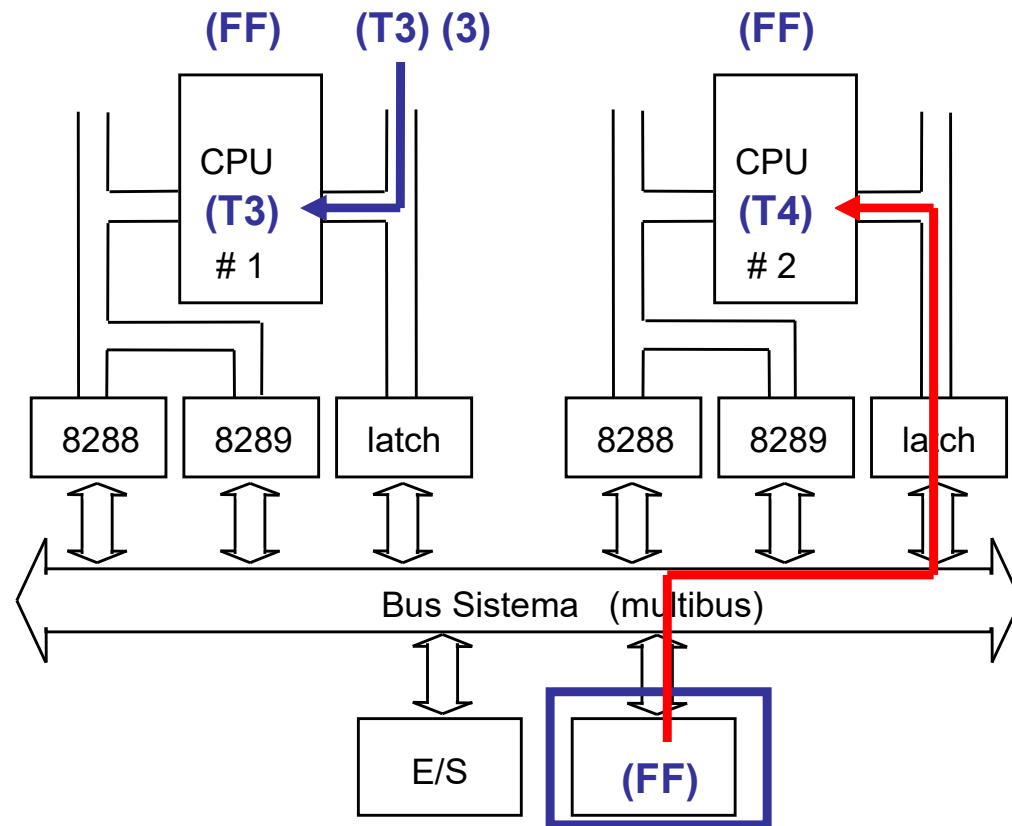
Tiempo = 13

FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (0) ▼
T4 (0) ▼
T5 (0) ▼

(bloqueada)

P1 está en su segundo ciclo de cálculo y P2 se prepara para leer el próximo valor de la tabla. A esta tarea le asignamos una unidad de tiempo.

Multiprocesamiento



Tiempo = 14

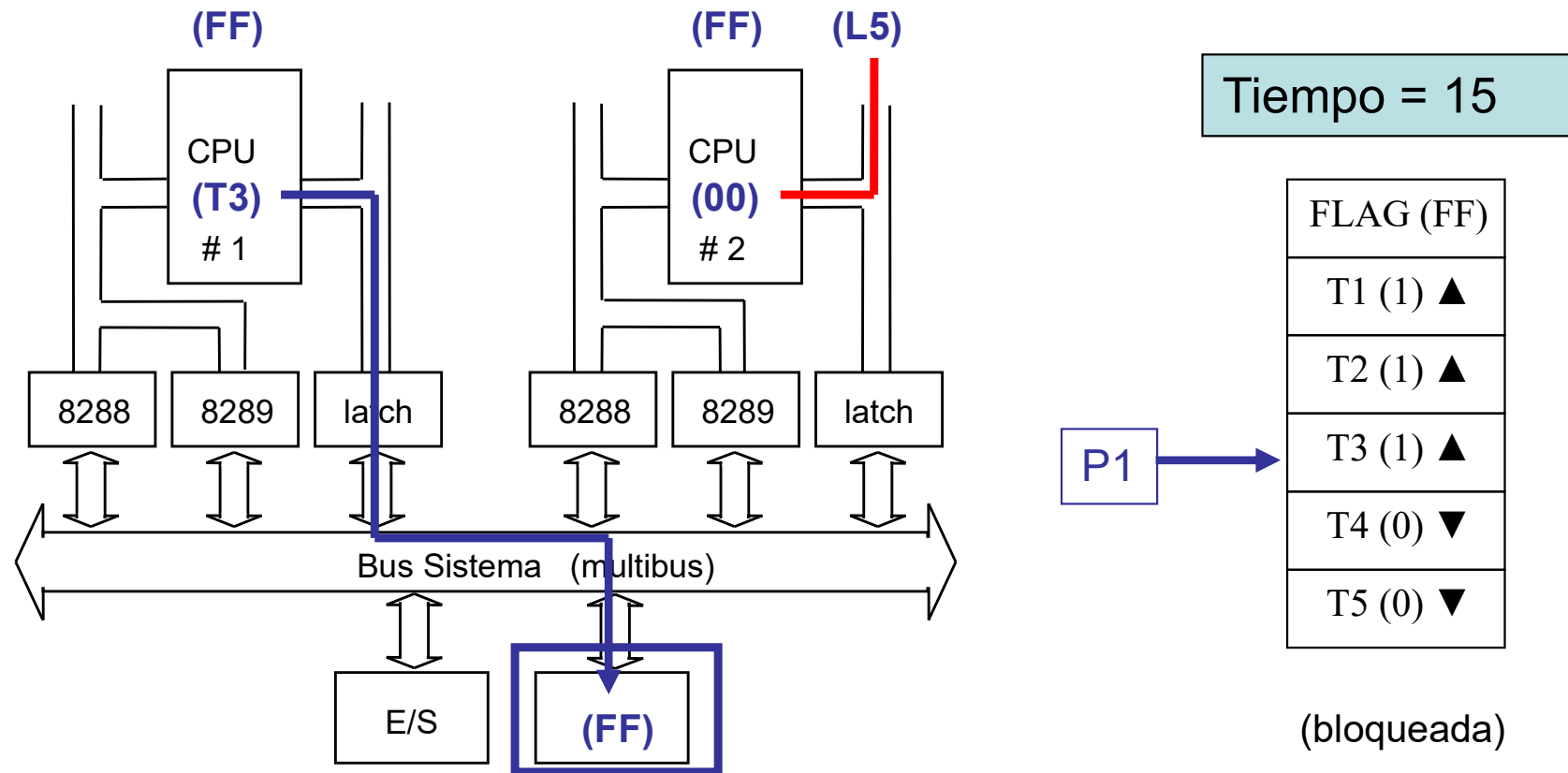
FLAG (00)
T1 (1) ▲
T2 (1) ▲
T3 (0) ▼
T4 (0) ▼
T5 (0) ▼

P2

(bloqueada)

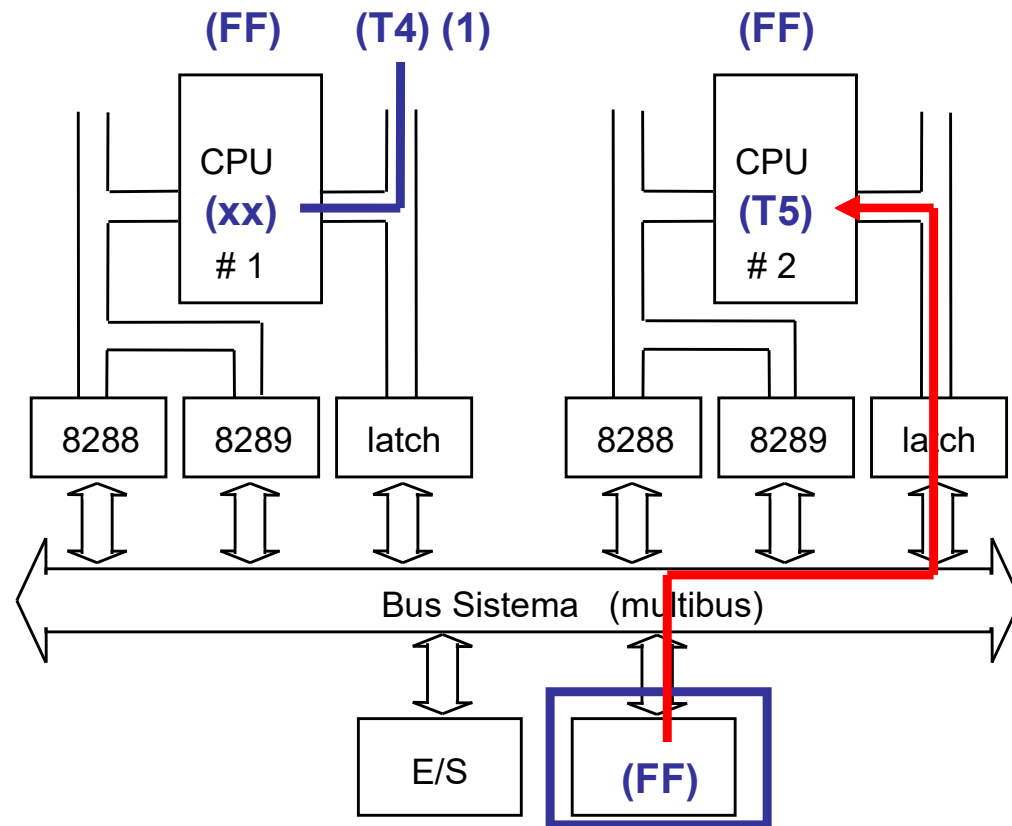
P1 realiza el cálculo de T3 y se dispone a escribirlo en la tabla. Supusimos que este cálculo demora 3 unidades de tiempo. Al mismo tiempo, P2 lee el valor de T3 desde la tabla. Observe que este dato no corresponde al mismo muestreo que el anterior.

Multiprocesamiento



P1 escribe el valor T3 en la tabla, mientras P2 está en su último paso de preparación para leer los valores de la tabla.

Multiprocesamiento



Tiempo = 16

FLAG (00)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (0) ▼
T5 (0) ▼

P2

(bloqueada)

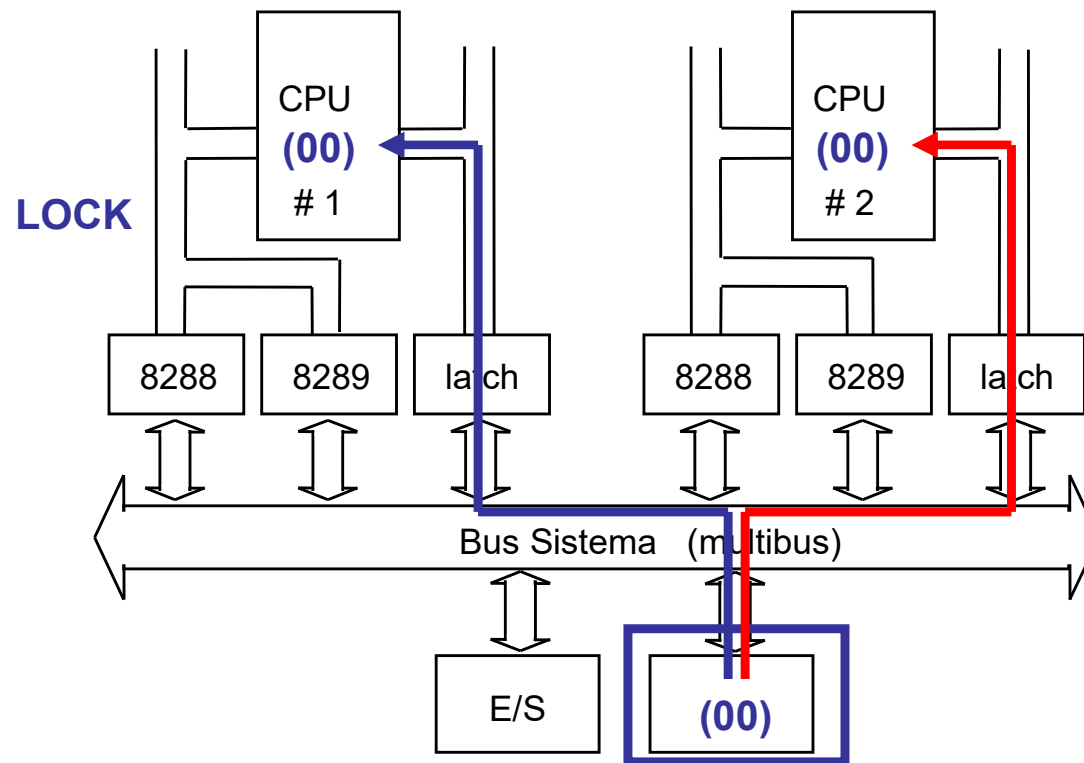
P1 realiza el cálculo de T4 y se dispone a escribirlo en la tabla. Supusimos que este cálculo demora 3 unidades de tiempo. Al mismo tiempo, P2 lee el valor de T5 desde la tabla. Observe que este dato no corresponde al mismo muestreo que el anterior.

Multiprocesamiento

P1	P2
T1 (1) ▲	T1 (1) ▲
T2 (1) ▲	T2 (0) ▼
T3 (1) ▲	T3 (0) ▼
T4 (0) ▼	T4 (0) ▼
T5 (0) ▼	T5 (0) ▼

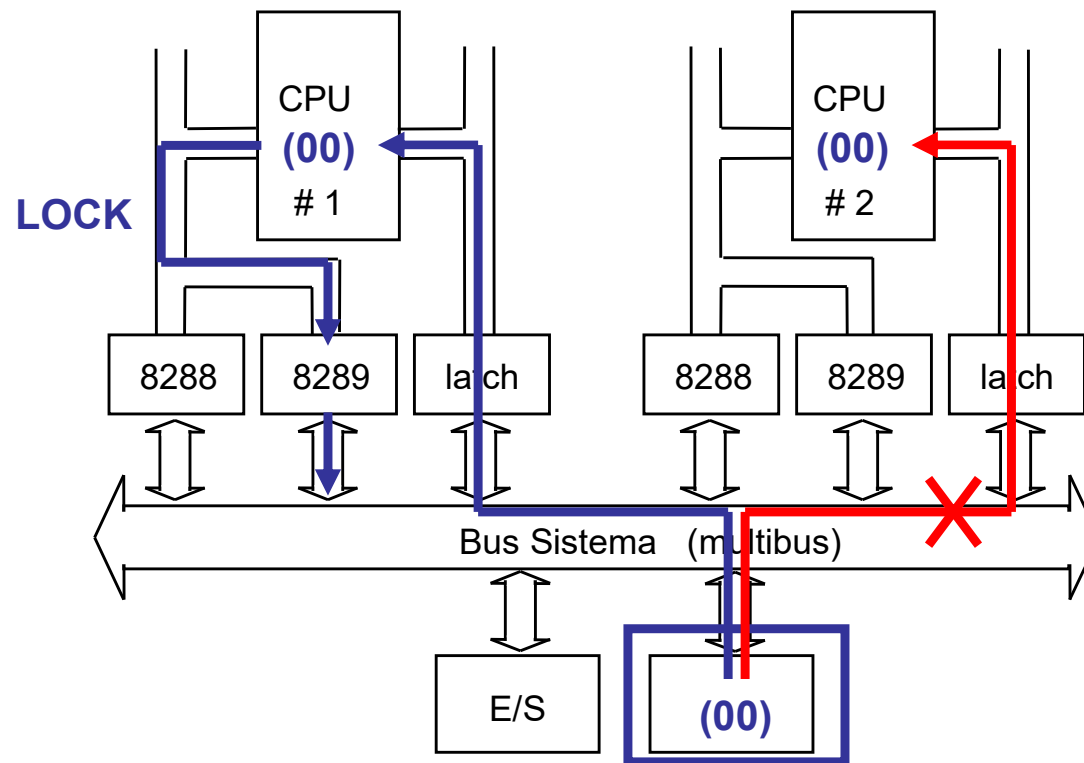
Como se puede apreciar, en el tiempo 16 el procesador P2 leyó los 5 valores, pero T1 corresponde a una muestra distinta (mas nueva) que los restantes (T2 a T5). Por otra parte P1 continúa actualizando la tabla. En este momento falta escribir dos valores.

Multiprocesamiento



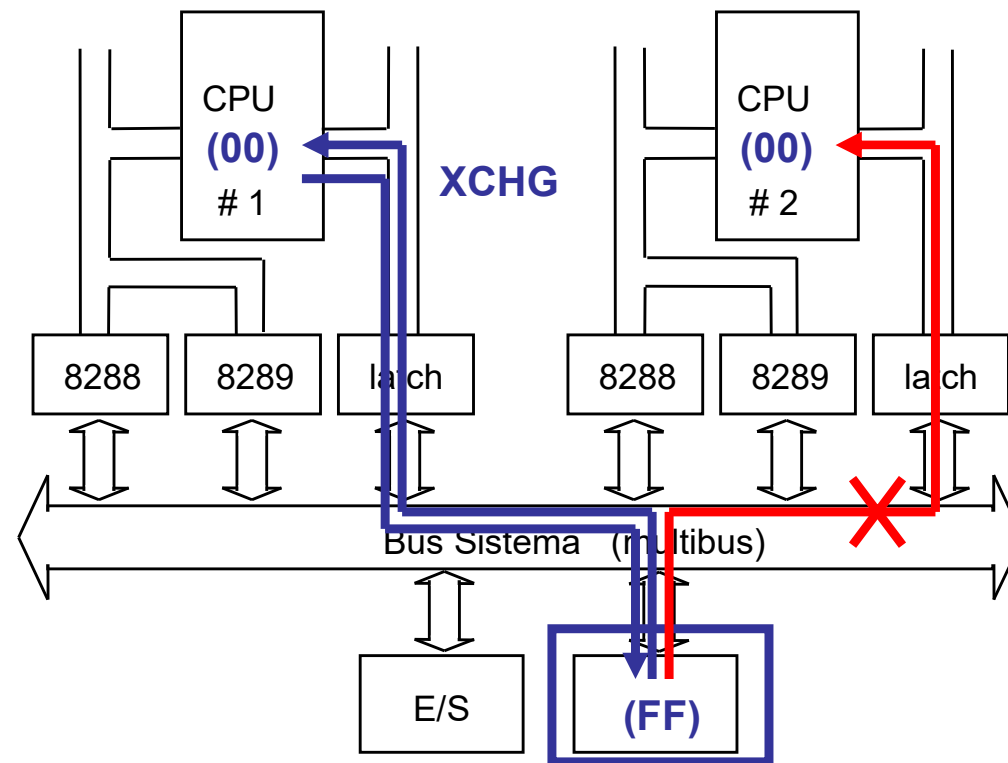
Recordemos que esto ocurrió porque los dos procesadores “vieron” el FLAG en 00 y por lo tanto consideraron que estaban habilitados para para operar con la tabla.

Multiprocesamiento



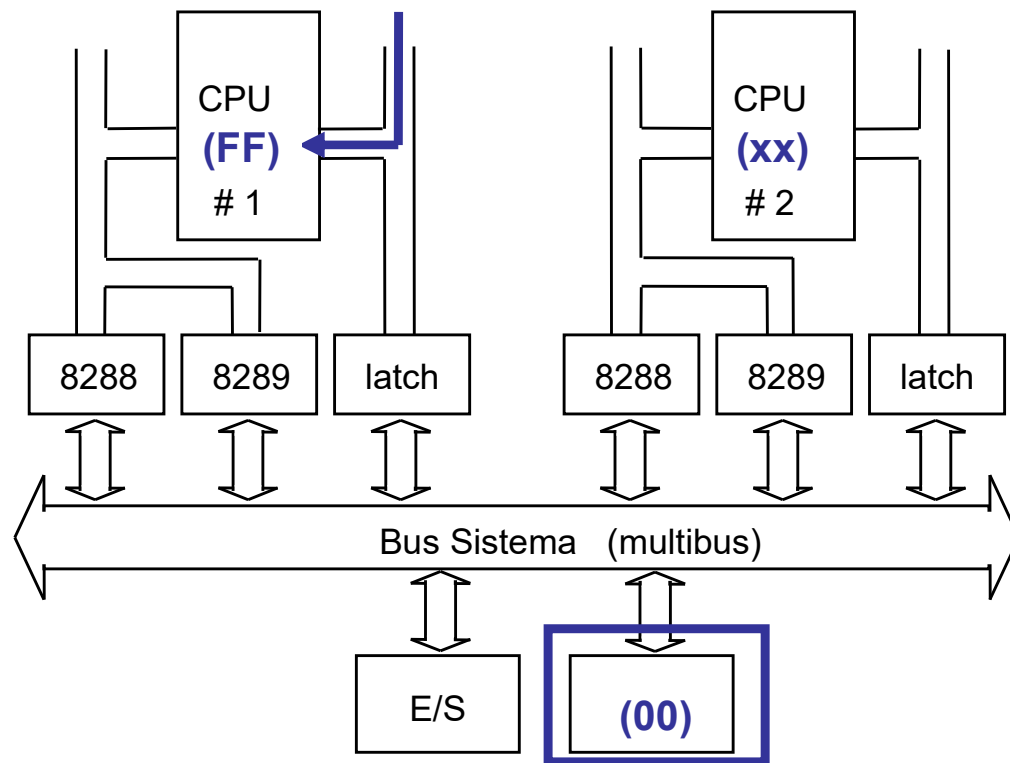
Los procesadores diseñados para trabajar en sistemas de procesadores múltiples cuentan con un prefijo (**instrucción LOCK**) para bloquear el bus con la **señal LOCK**. Mientras el bus está bloqueado por P1, P2 no puede acceder al mismo.

Multiprocesamiento



Los procesadores diseñados para trabajar en sistemas de procesadores múltiples cuentan con una **instrucción XCHG** para intercambiar un registro y memoria. Esta instrucción requiere dos ciclos de bus. Adicionándole el prefijo LOCK, se puede verificar la bandera y al mismo tiempo bloquearla, para que el otro procesador vea el estado correcto y pueda actuar en consecuencia. Veremos la secuencia nuevamente.

Multiprocesamiento



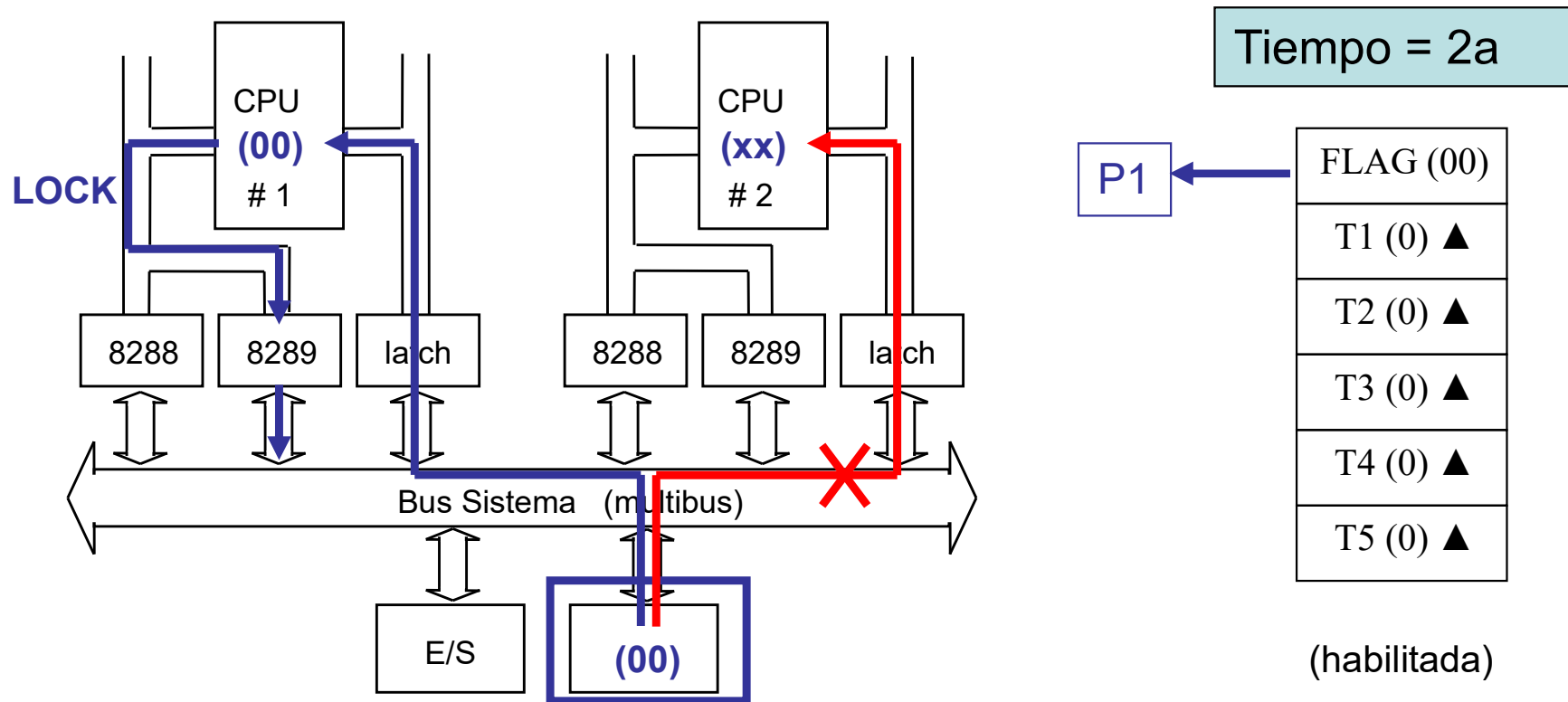
Tiempo = 1

FLAG (00)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

(habilitada)

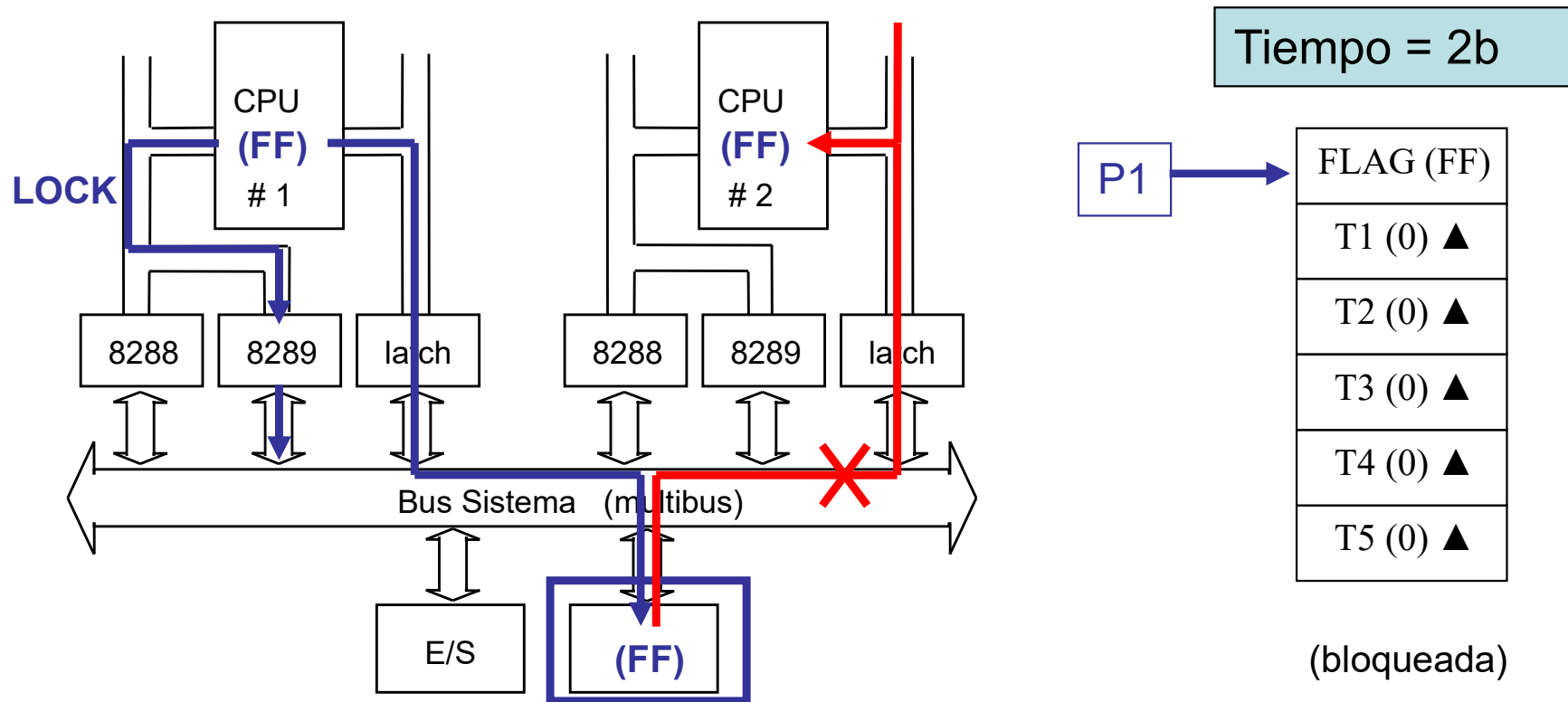
Haremos la secuencia paso a paso, avanzando con cada ciclo de reloj. El estado inicial es la tabla habilitada, aunque para esta forma de trabajo es indiferente.
P1 carga FF en el registro A para luego hacer LOCK XCHG.

Multiprocesamiento



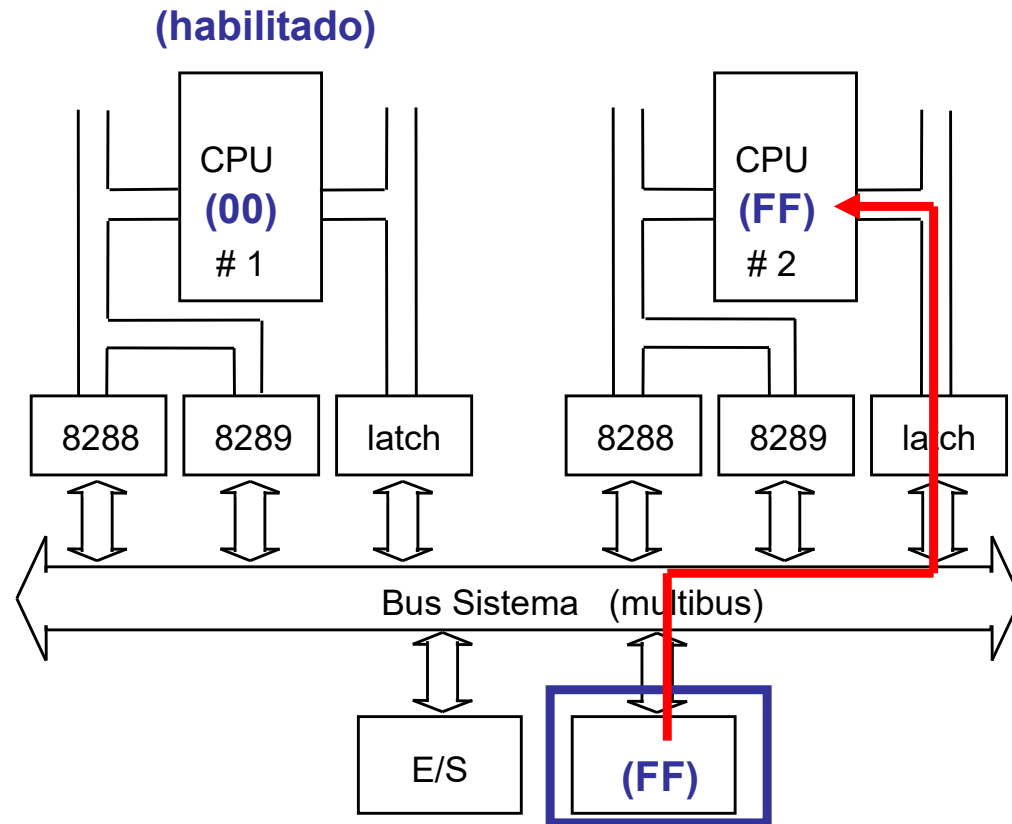
P1 hace LOCK XCHG. Esta instrucción, en el primer ciclo de bus lee el FLAG y lo deposita en un registro temporario del procesador.

Multiprocesamiento



En el segundo ciclo, (recordemos que el bus está bloqueado) escribe FF (que estaba en el registro A) en el FLAG de la tabla. Luego copia el temporario en A, para poder operar con su valor. P2 carga FF en A.

Multiprocesamiento



Tiempo = 3

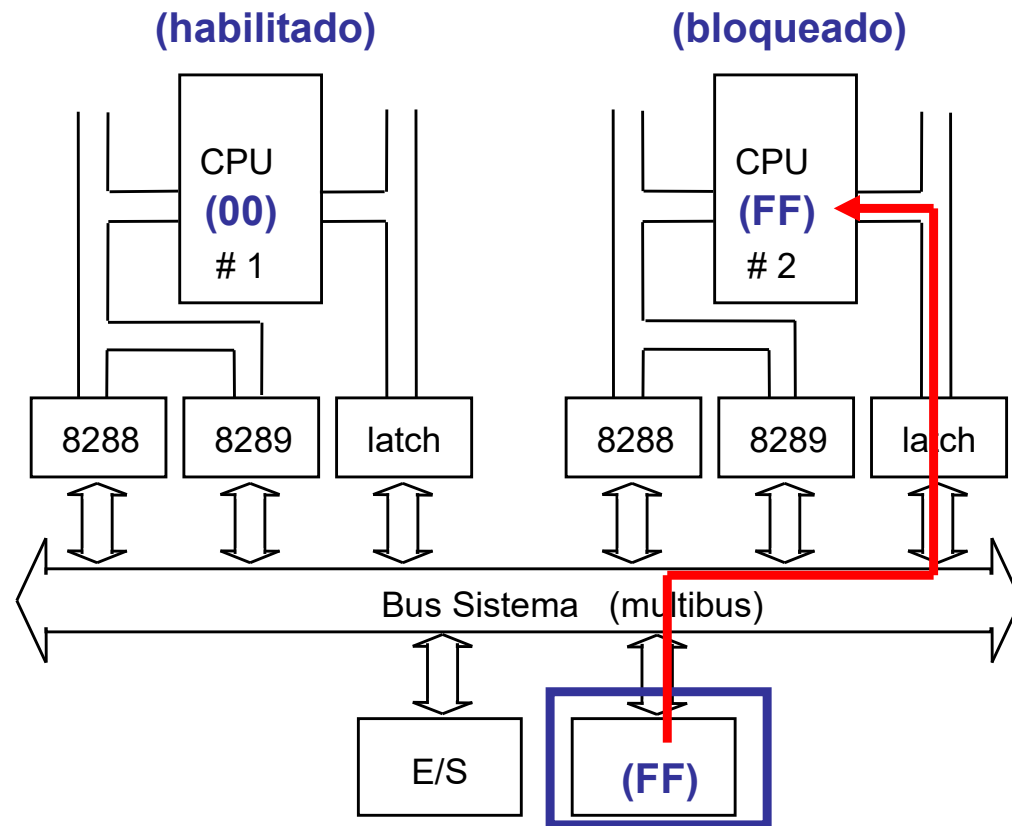
FLAG (FF)	→ P2
T1 (0) ▲	
T2 (0) ▲	
T3 (0) ▲	
T4 (0) ▲	
T5 (0) ▲	

(bloqueada)

El bus está desbloqueado, pero ya no es problema porque la tabla está bloqueada por el FLAG en FF.

P2 lee el FLAG con LOCK XCHG (de la misma forma que lo había hecho P1).

Multiprocesamiento



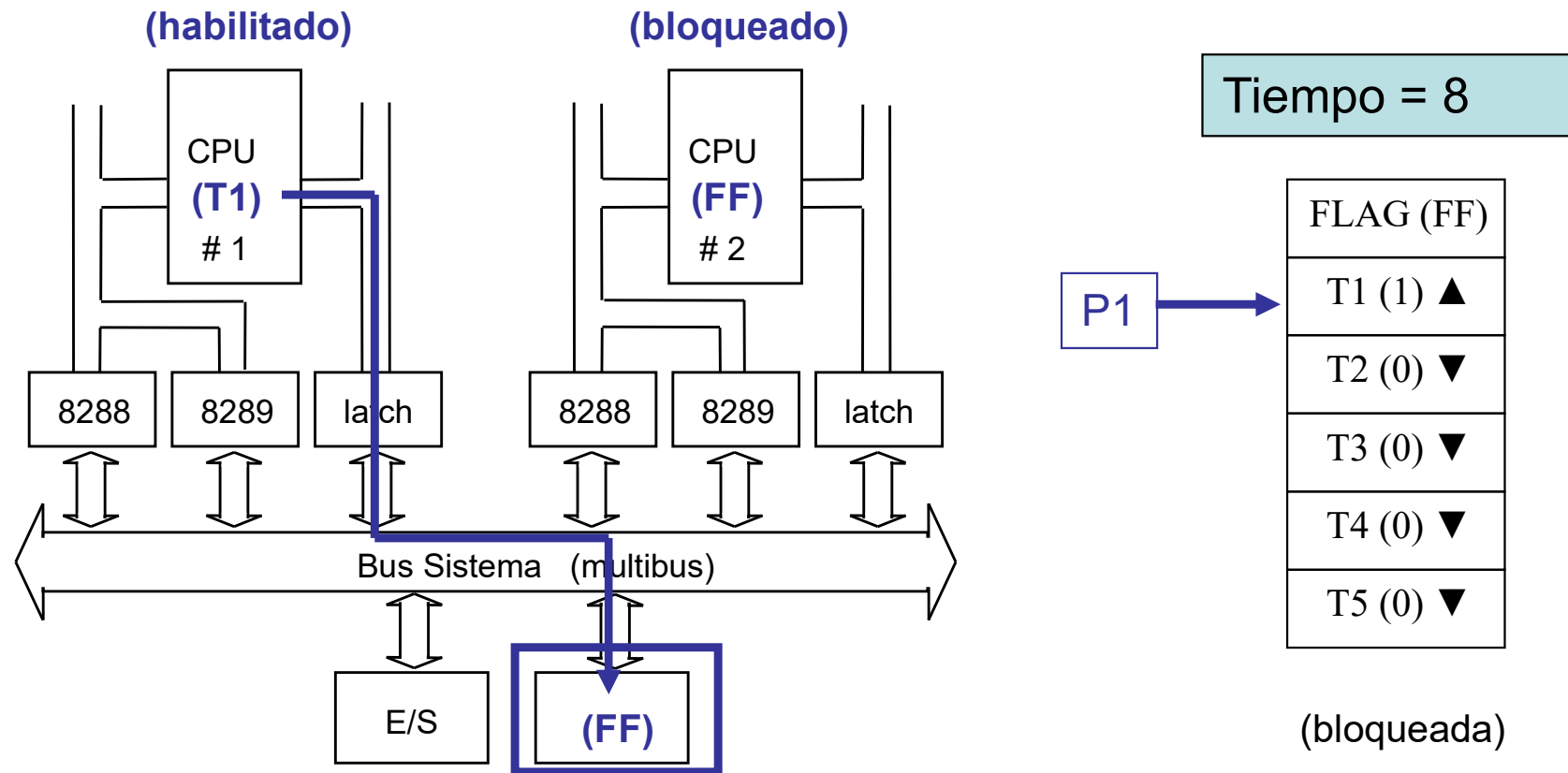
Tiempo = 4

FLAG (FF)
T1 (0) ▲
T2 (0) ▲
T3 (0) ▲
T4 (0) ▲
T5 (0) ▲

(bloqueada)

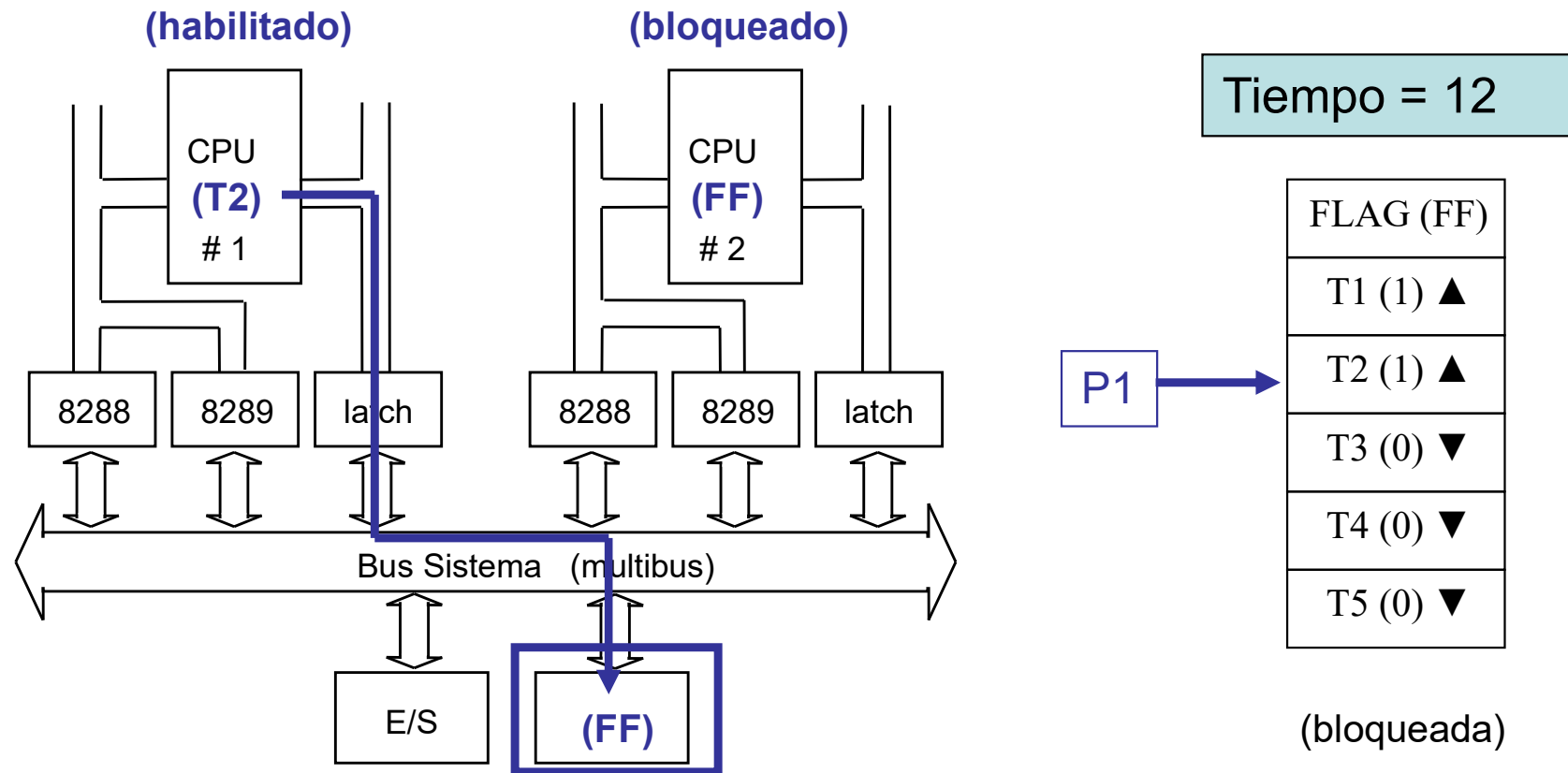
Con TEST AL,AL verifica que no puede acceder a la tabla porque FLAG está en FF.

Multiprocesamiento



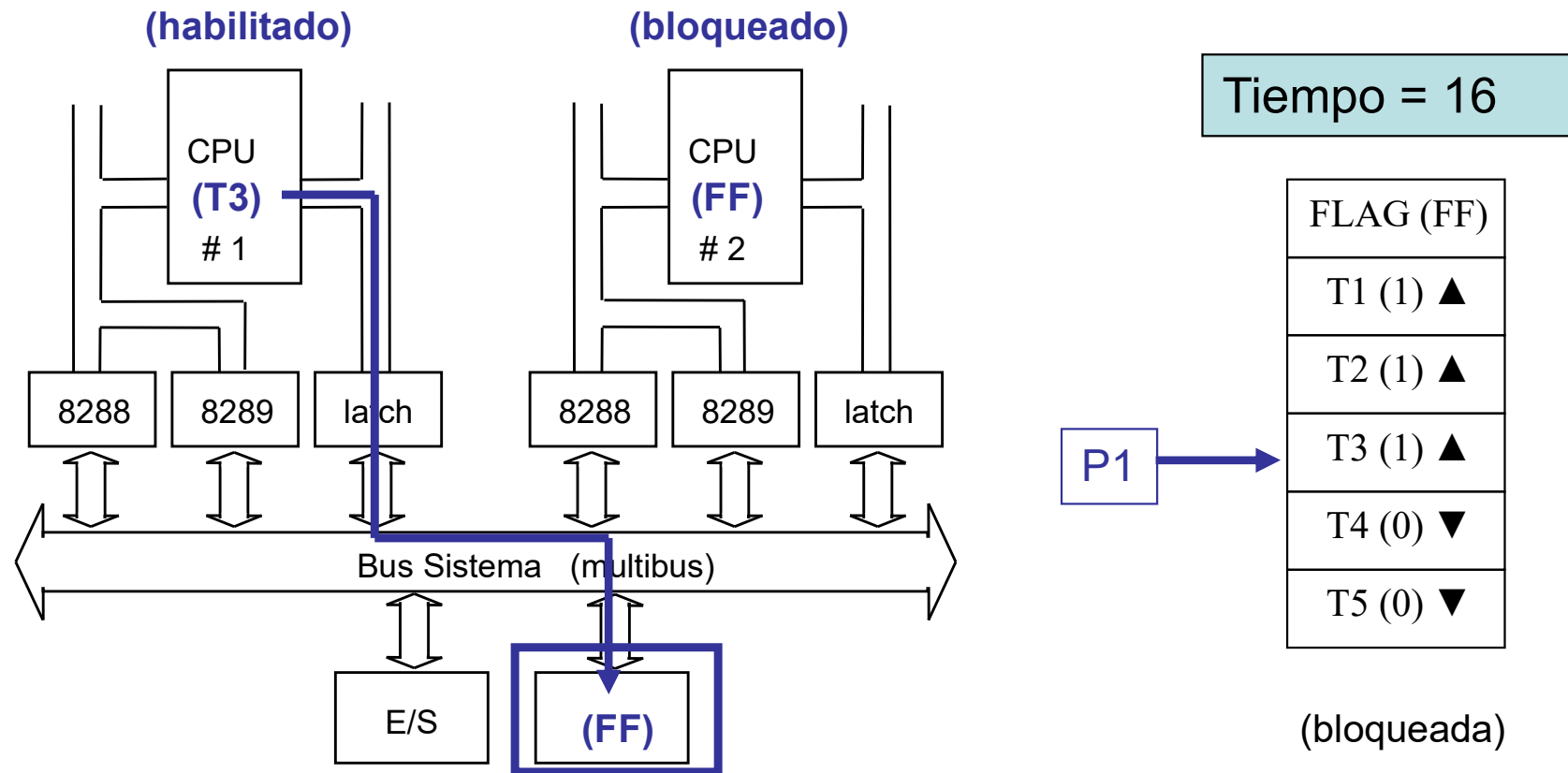
P1 escribe T1 en la tabla. P2 permanece bloqueado por FLAG = FF.

Multiprocesamiento



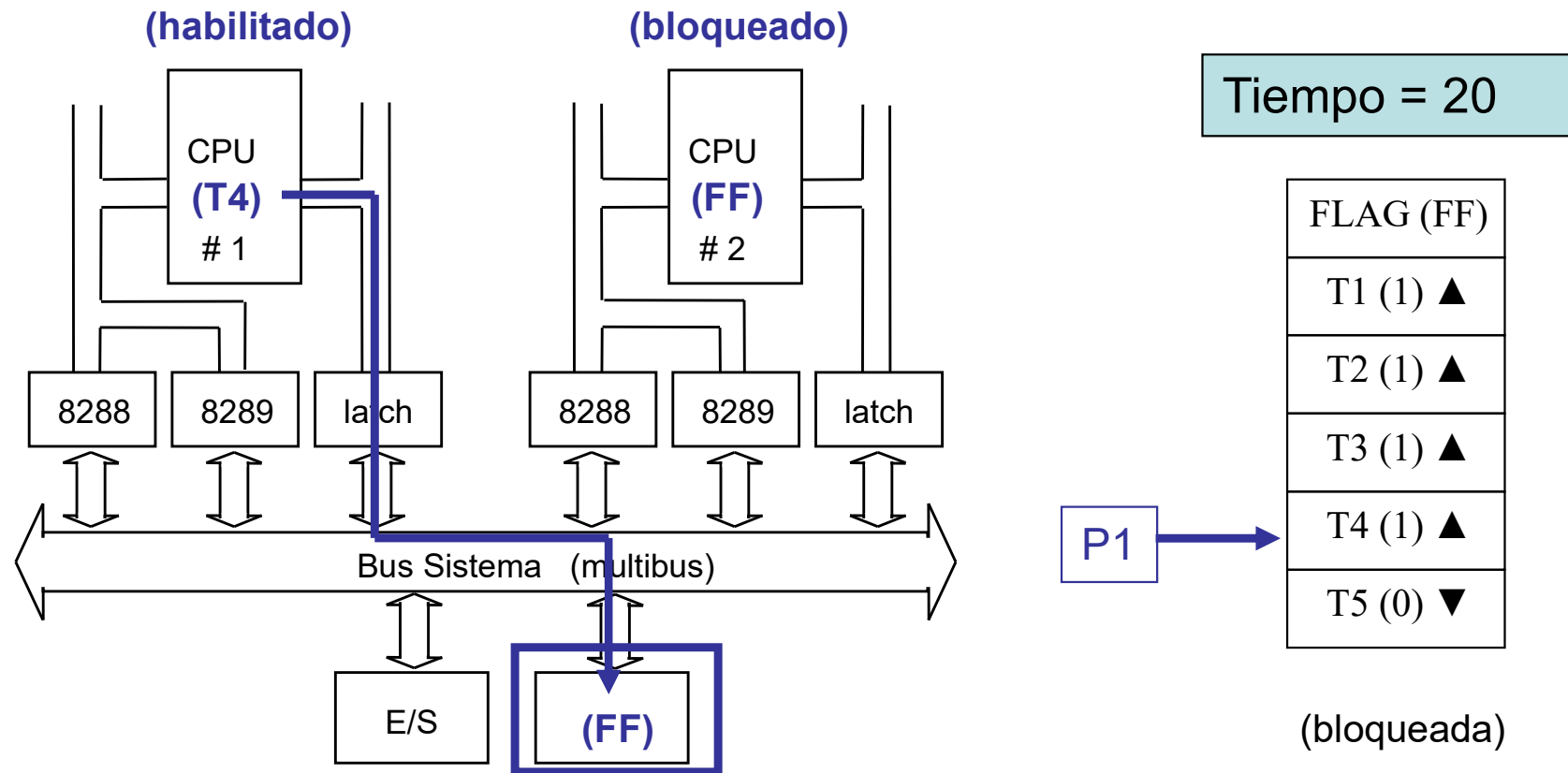
P1 escribe T2 en la tabla. P2 permanece bloqueado por FLAG = FF.

Multiprocesamiento



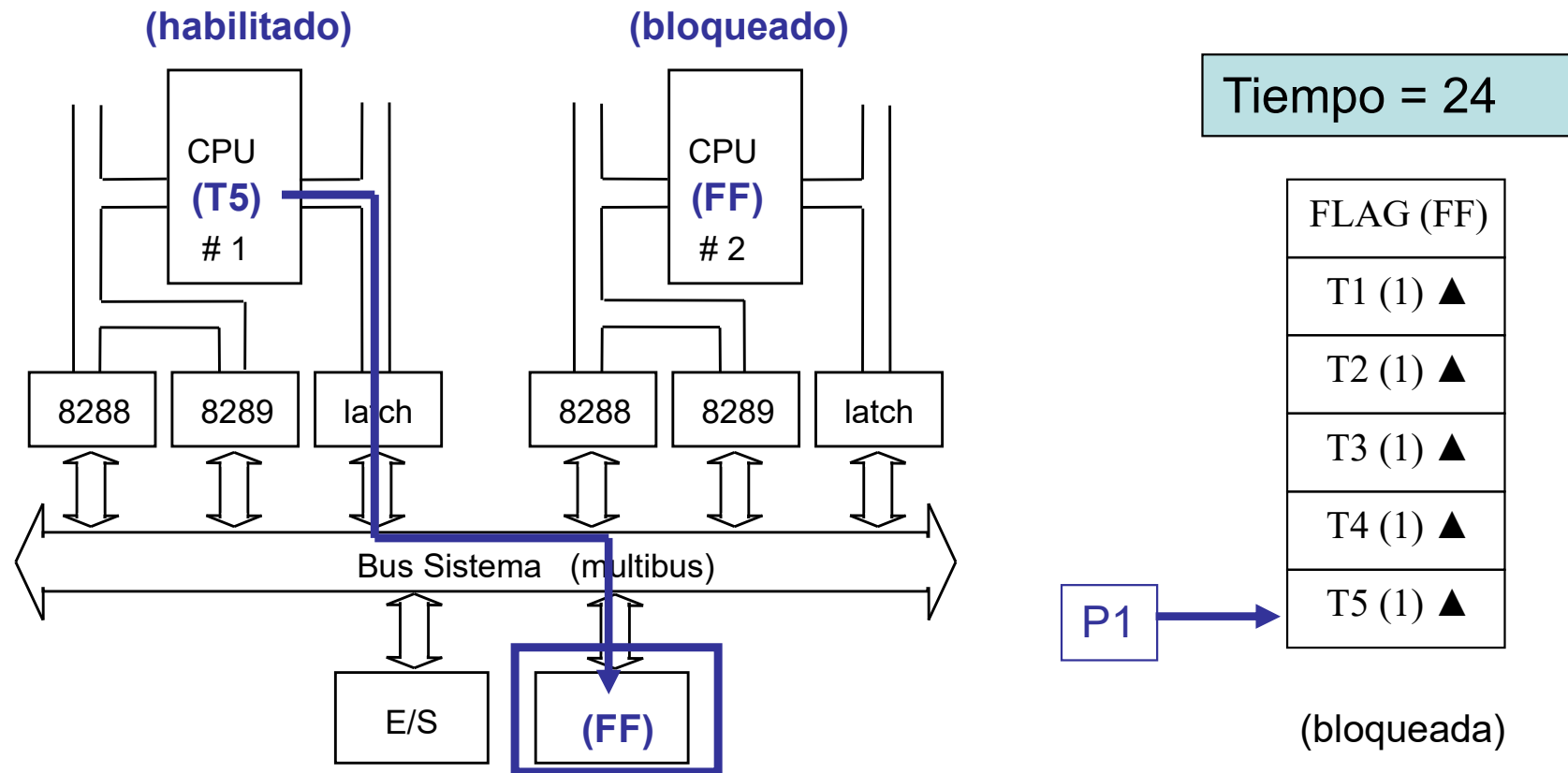
P1 escribe T3 en la tabla. P2 permanece bloqueado por FLAG = FF.

Multiprocesamiento



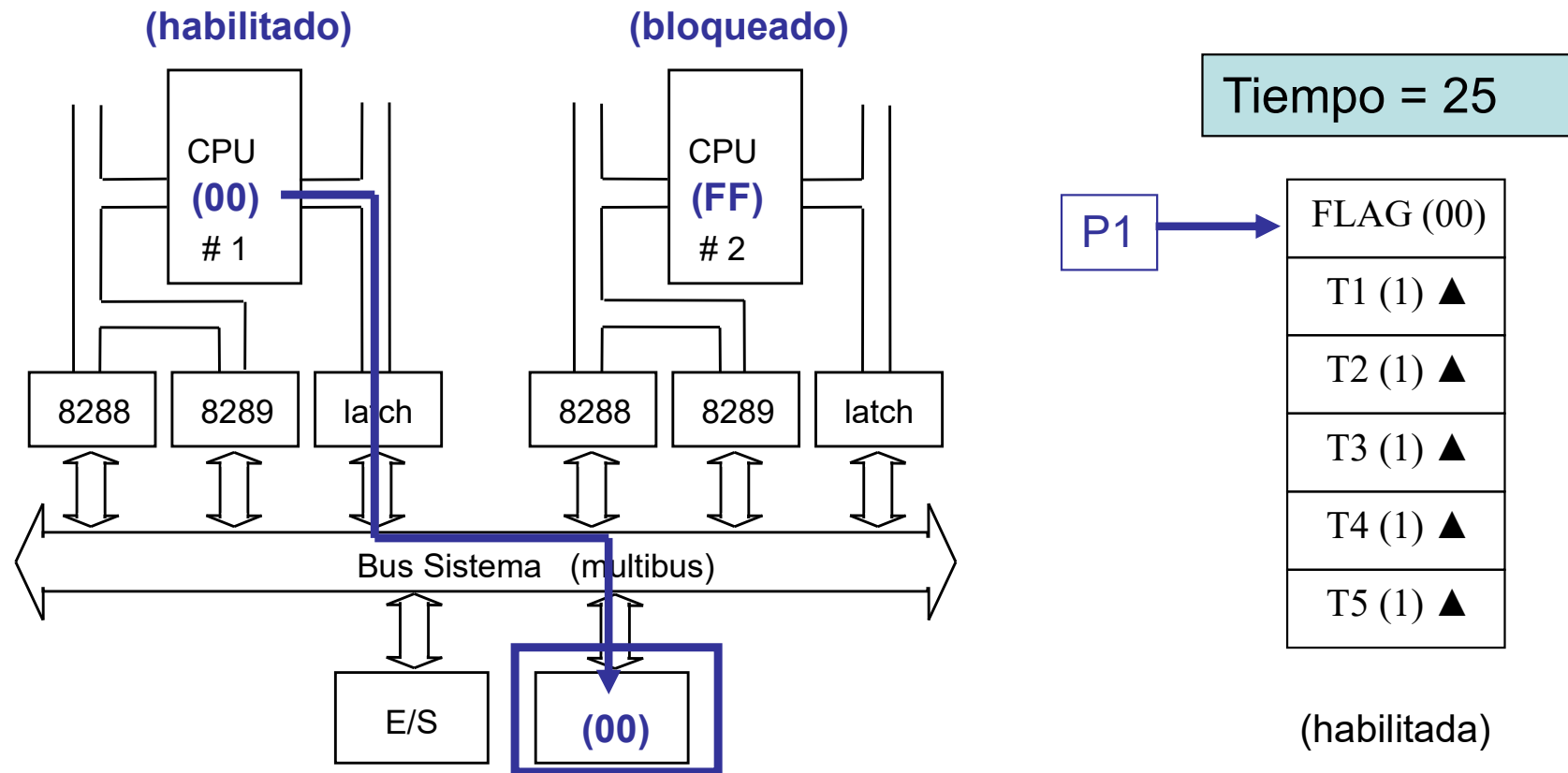
P1 escribe T4 en la tabla. P2 permanece bloqueado por FLAG = FF.

Multiprocesamiento



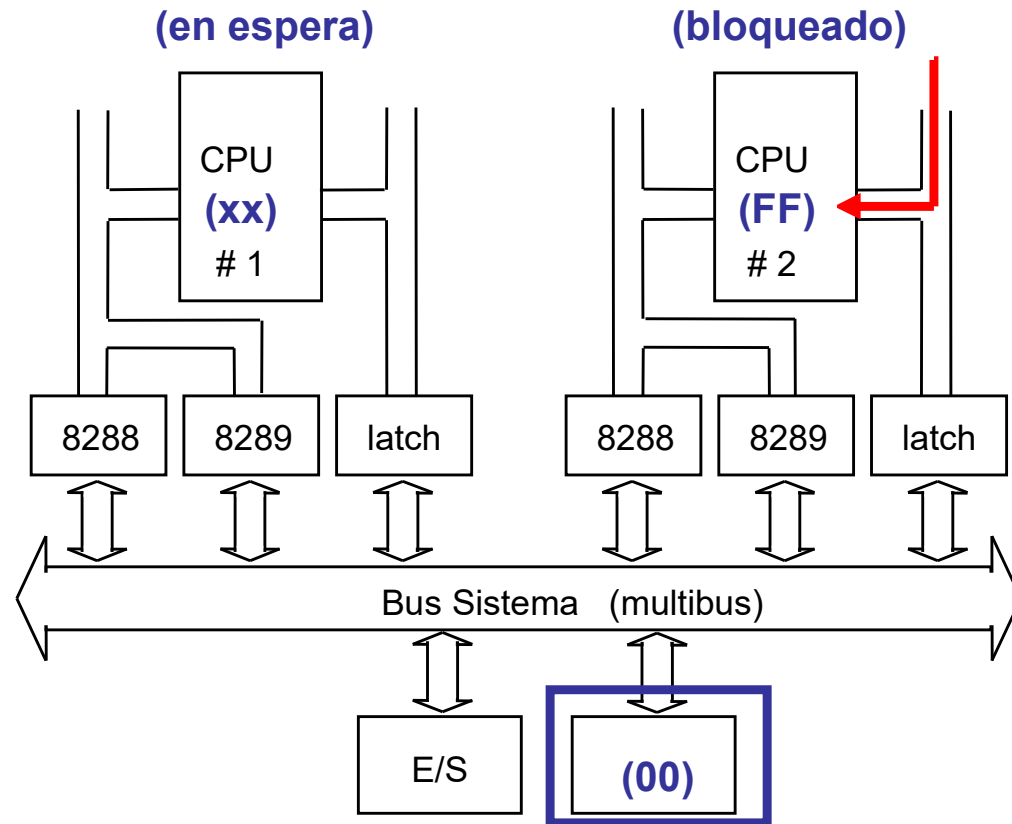
P1 escribe T5 en la tabla. P2 permanece bloqueado por FLAG = FF.

Multiprocesamiento



P1 escribe 00 en FLAG, ya que terminó de actualizar los cinco valores.

Multiprocesamiento



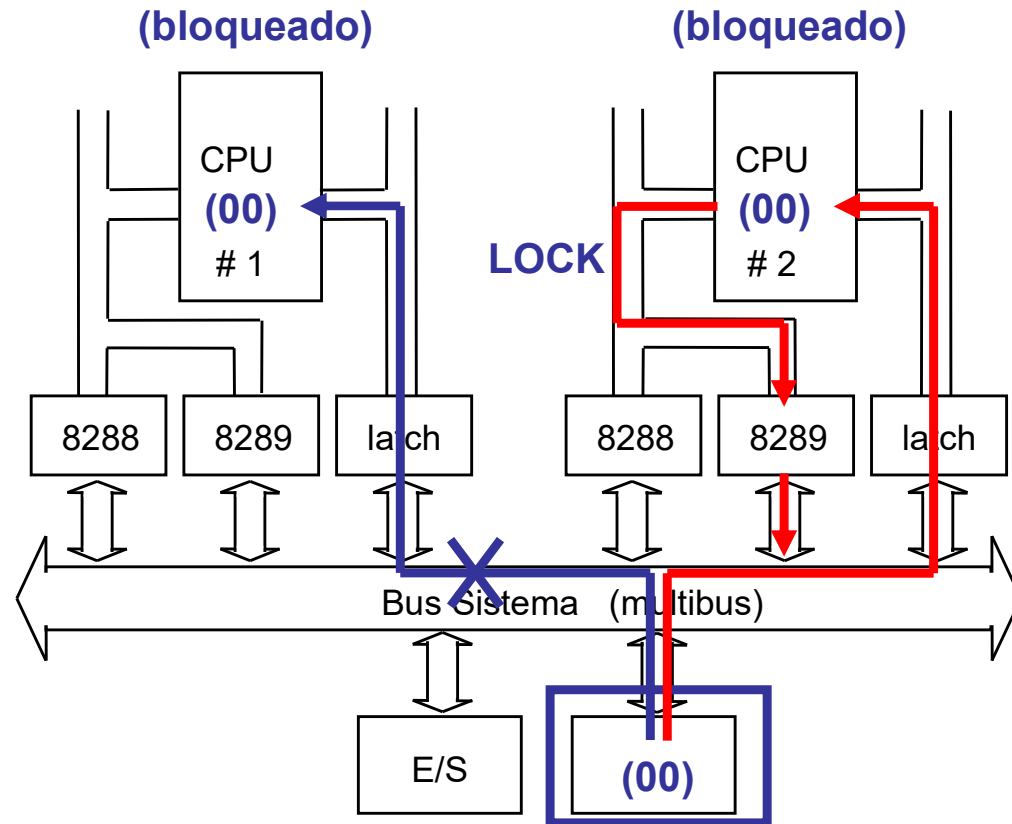
Tiempo = 26

FLAG (00)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

(habilitada)

P2 intenta verificar si puede acceder a la tabla. Carga FF en el registro A para luego hacer LOCK XCHG.

Multiprocesamiento



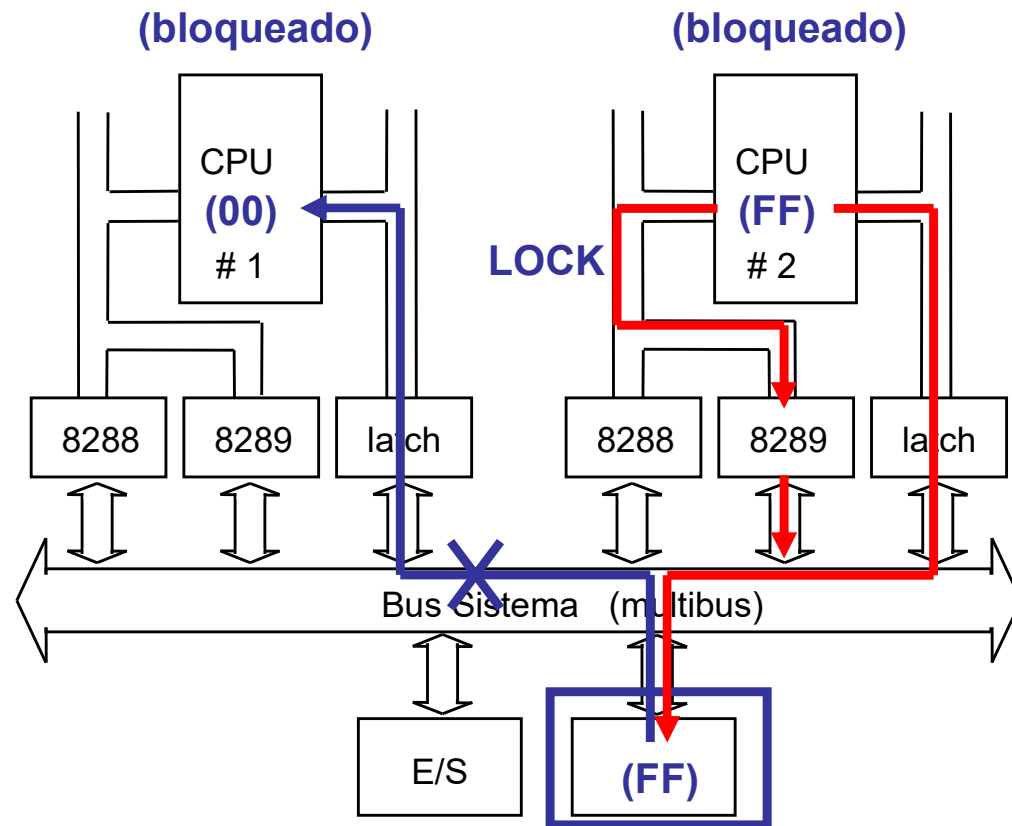
Tiempo = 27a

FLAG (00)	→ P2
T1 (1) ▲	
T2 (1) ▲	
T3 (1) ▲	
T4 (1) ▲	
T5 (1) ▲	

(habilitada)

P2 hace LOCK XCHG. Esta instrucción, en el primer ciclo de bus lee el FLAG y lo deposita en un registro temporario del procesador.

Multiprocesamiento



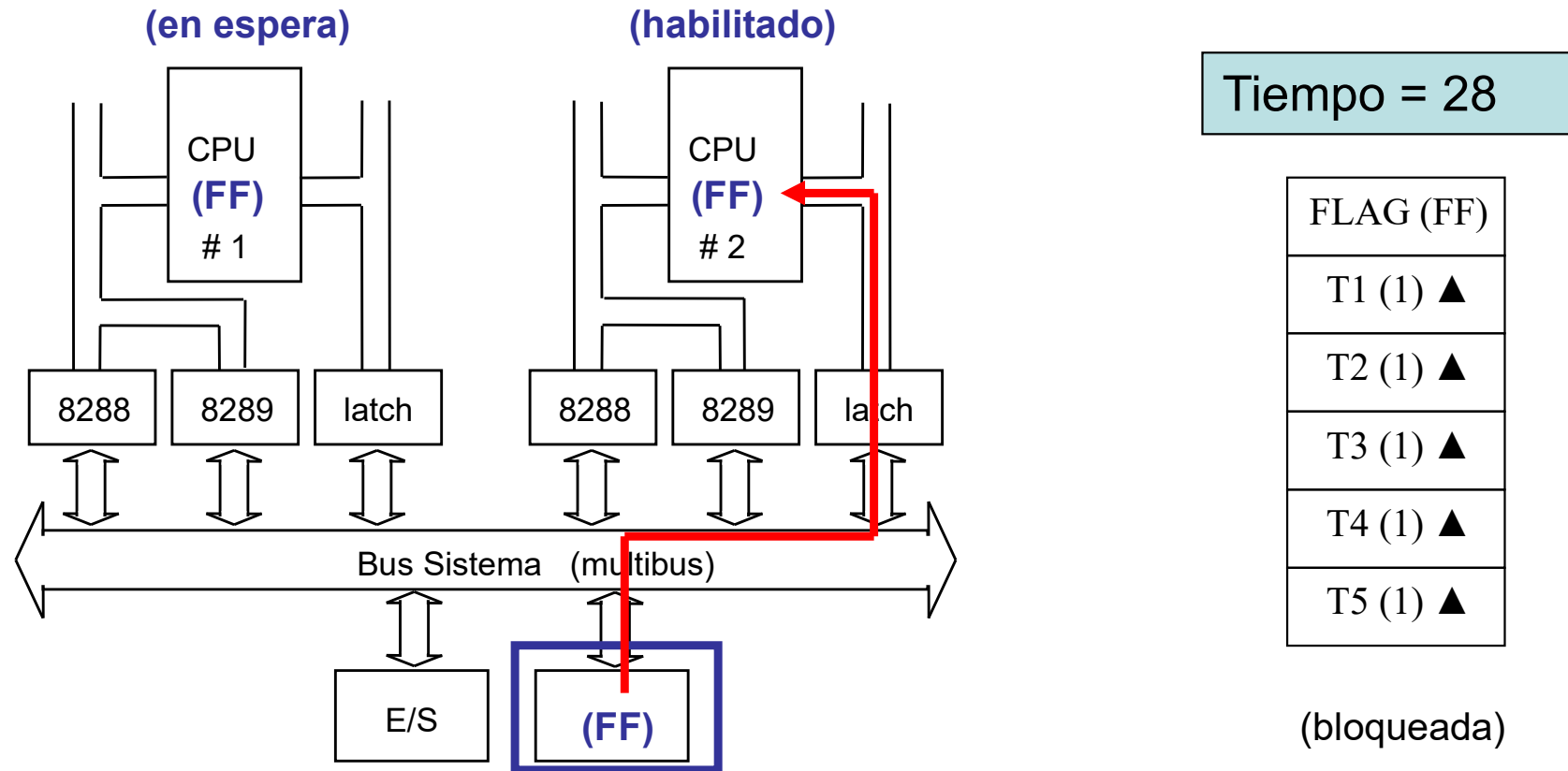
Tiempo = 27b

FLAG (FF)	← P2
T1 (1) ▲	
T2 (1) ▲	
T3 (1) ▲	
T4 (1) ▲	
T5 (1) ▲	

(bloqueada)

En el segundo ciclo, (recordemos que el bus está bloqueado) escribe FF (que estaba en el registro A) en el FLAG de la tabla. Luego copia el temporario en A, para poder operar con su valor.

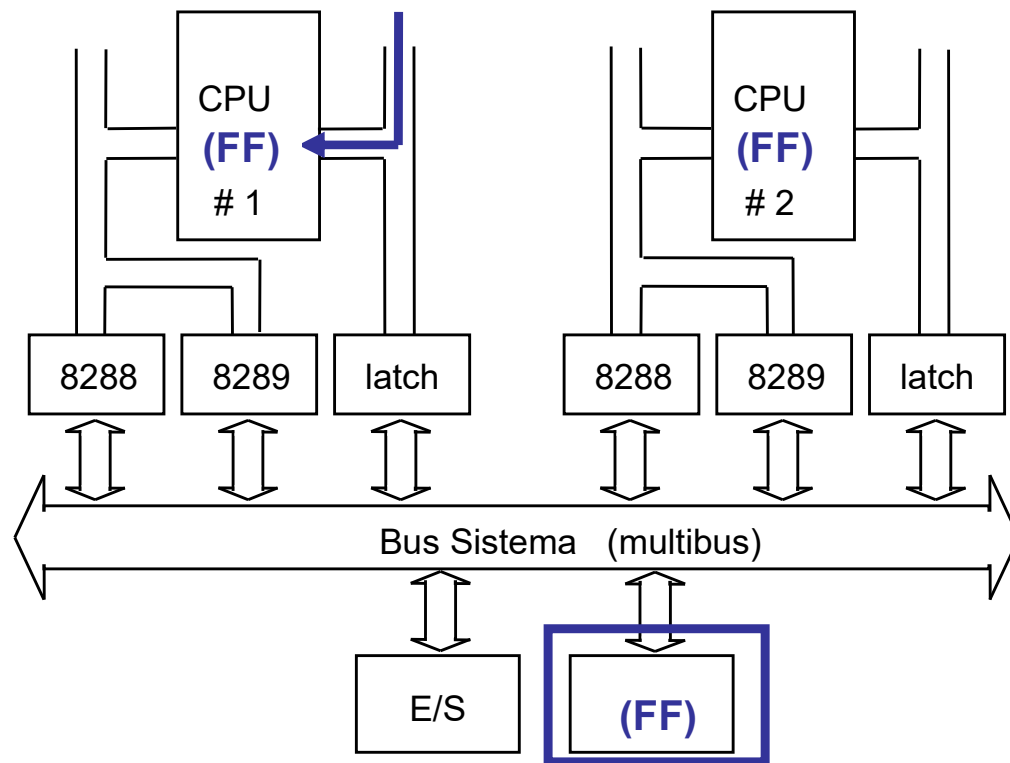
Multiprocesamiento



Con TEST AL,AL, P2 verifica que puede acceder a la tabla porque FLAG está en 00. En FLAG quedó el FF que puso XCHG, lo que inhabilitará a P1 cuando intente acceder.

Multiprocesamiento

(habilitado)



Tiempo = 29

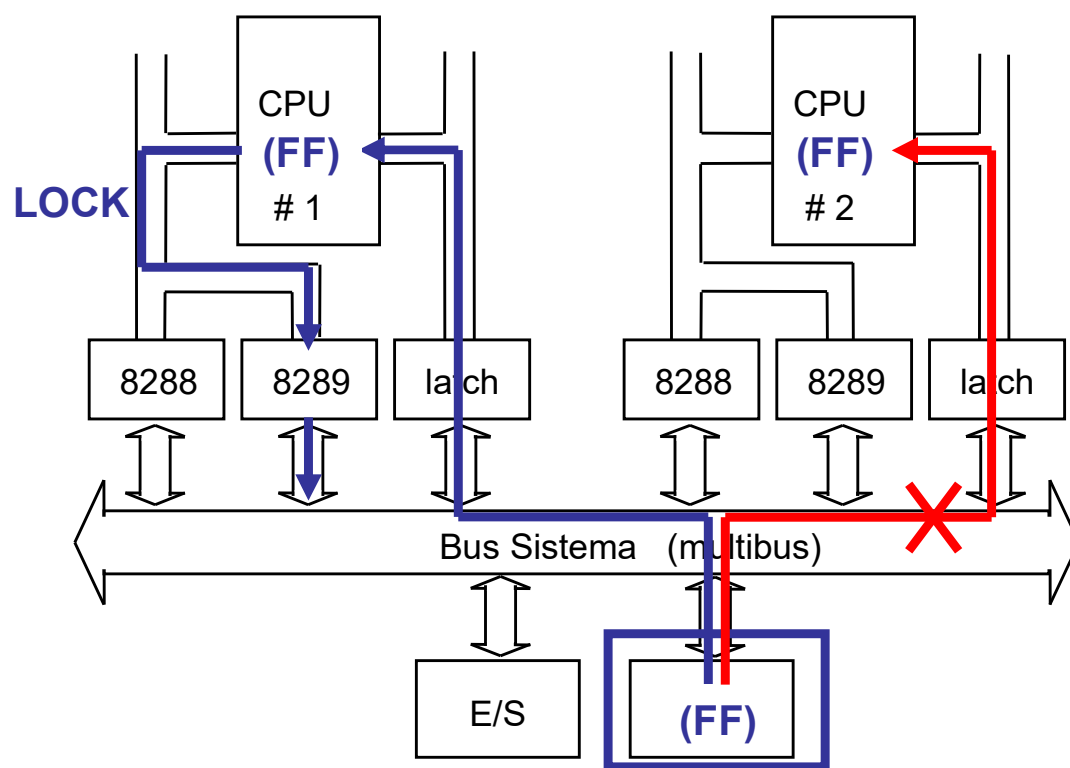
FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

(bloqueada)

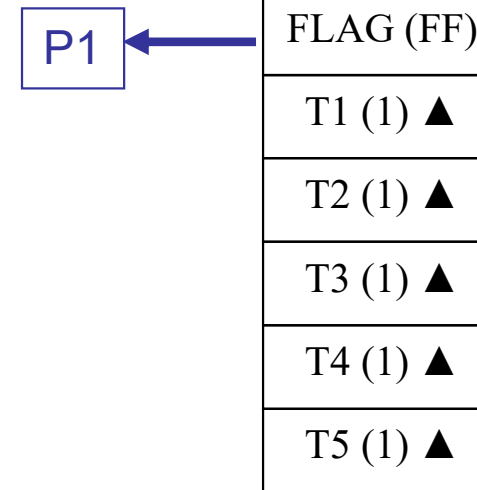
En un intento por escribir en la tabla, P1 carga FF en el registro A para luego hacer LOCK XCHG.

Multiprocesamiento

(habilitado)



Tiempo = 30a

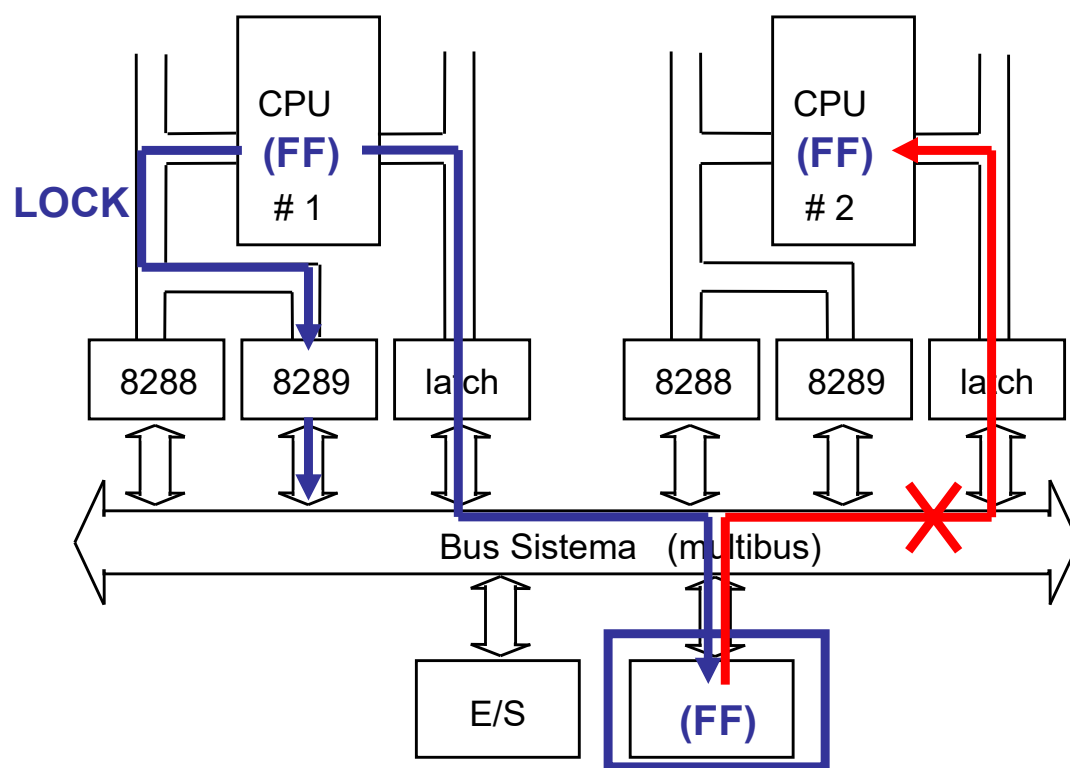


(bloqueada)

P1 hace LOCK XCHG. Esta instrucción, en el primer ciclo de bus lee el FLAG y lo deposita en un registro temporario del procesador.

Multiprocesamiento

(habilitado)



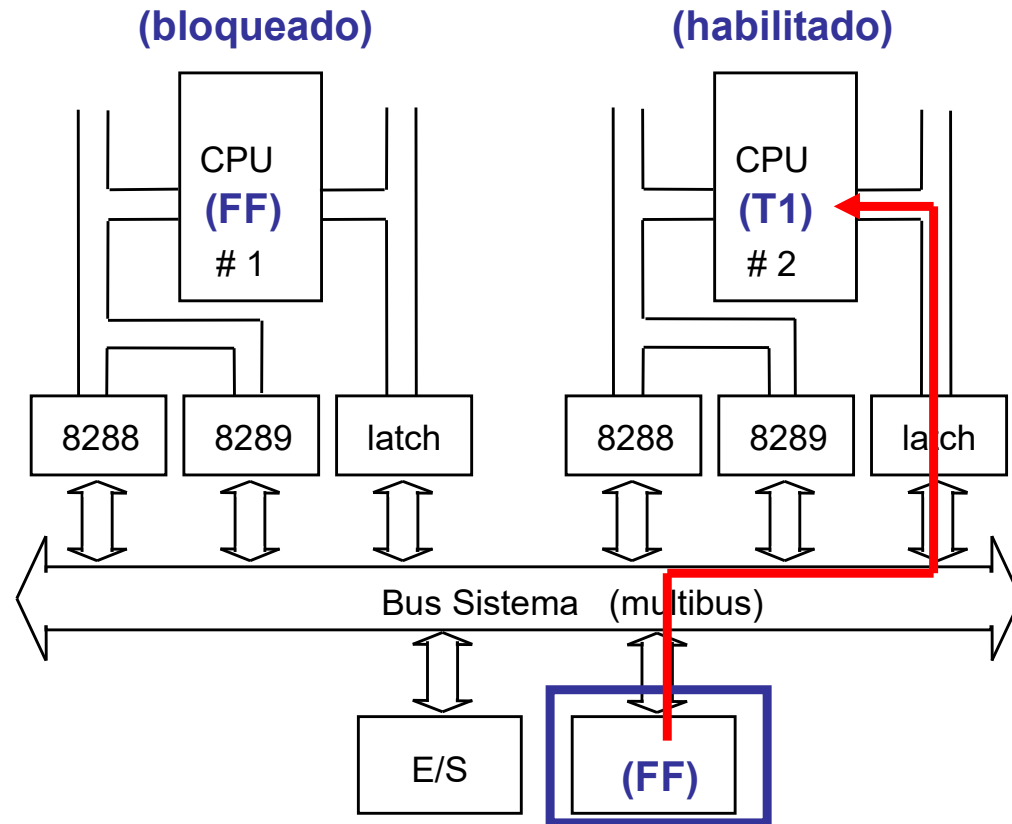
Tiempo = 30b

P1	FLAG (FF)
	T1 (1) ▲
	T2 (1) ▲
	T3 (1) ▲
	T4 (1) ▲
	T5 (1) ▲

(bloqueada)

En el segundo ciclo, (recordemos que el bus está bloqueado) escribe FF (que estaba en el registro A) en el FLAG de la tabla. Luego copia el temporario en A, para poder operar con su valor.

Multiprocesamiento



Tiempo = 31

FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

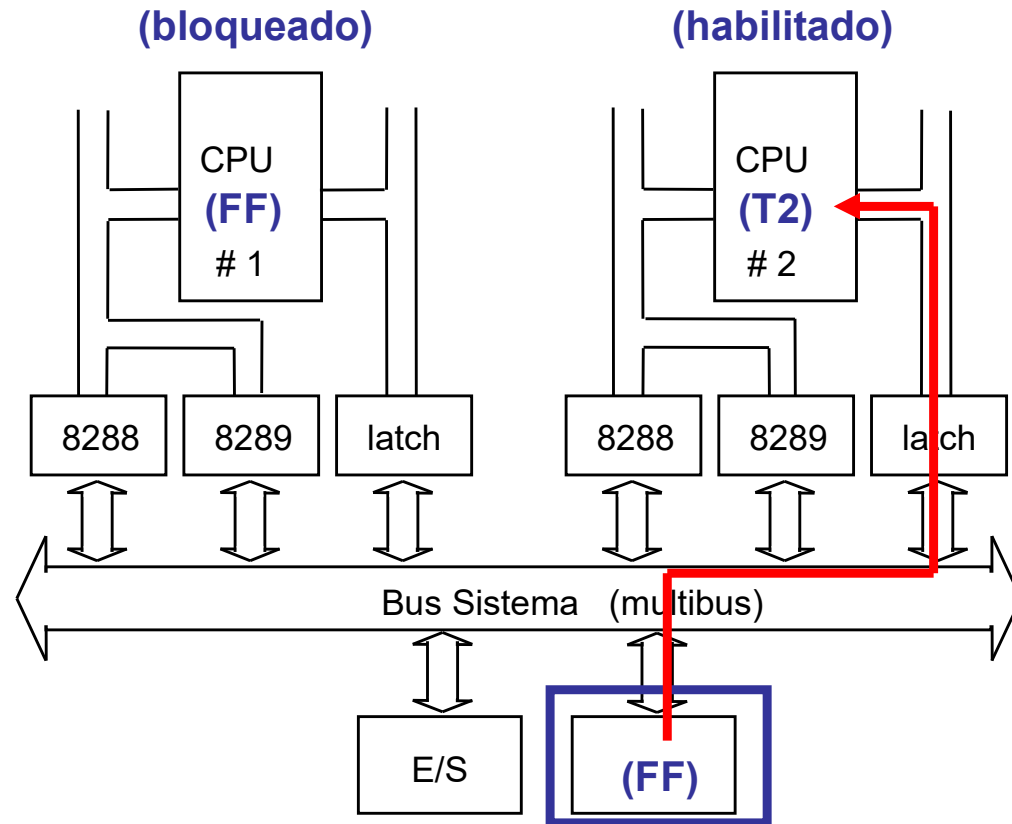
P2

(bloqueada)

P1 hace TEST AL,AL, verificando que FLAG estaba en FF y por ende queda bloqueado para escribir.

P2 lee el primer valor de la tabla.

Multiprocesamiento



Tiempo = 33

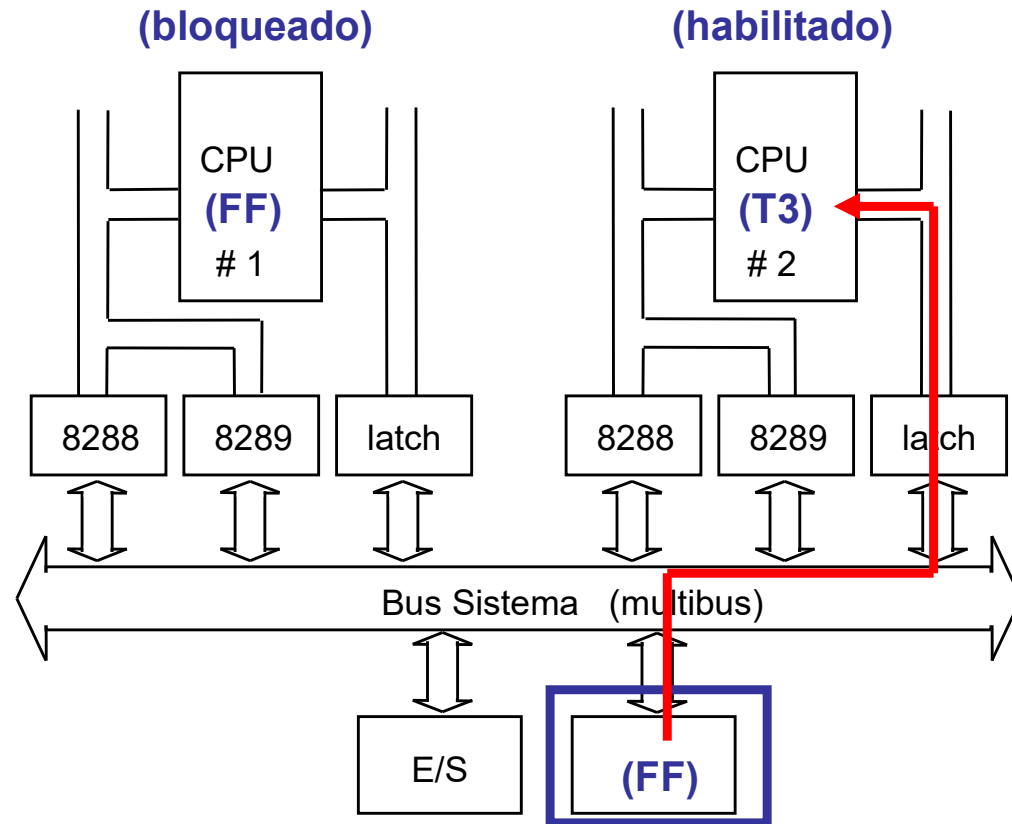
FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

P2

(bloqueada)

P2 lee T2. P1 puede intentar escribir, pero encontrará FF en FLAG y deberá esperar. Esto no afectará las posteriores lecturas de P2 sobre la tabla.

Multiprocesamiento



Tiempo = 35

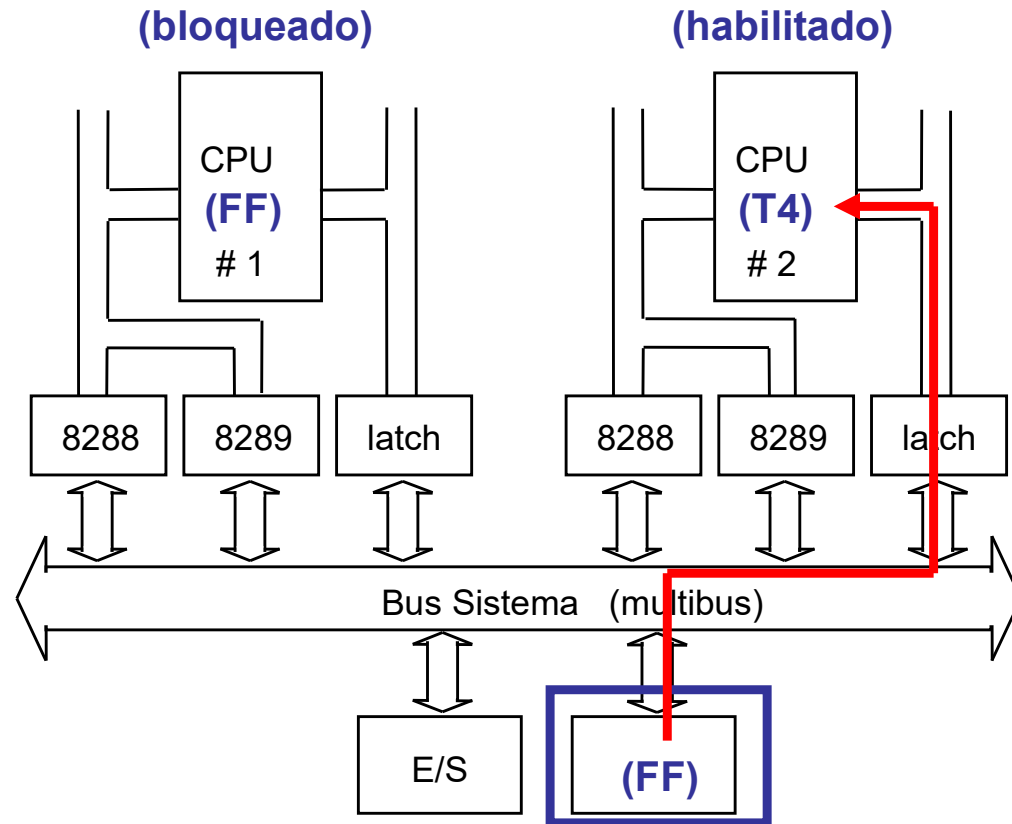
FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

P2

(bloqueada)

P2 lee T3

Multiprocesamiento



Tiempo = 37

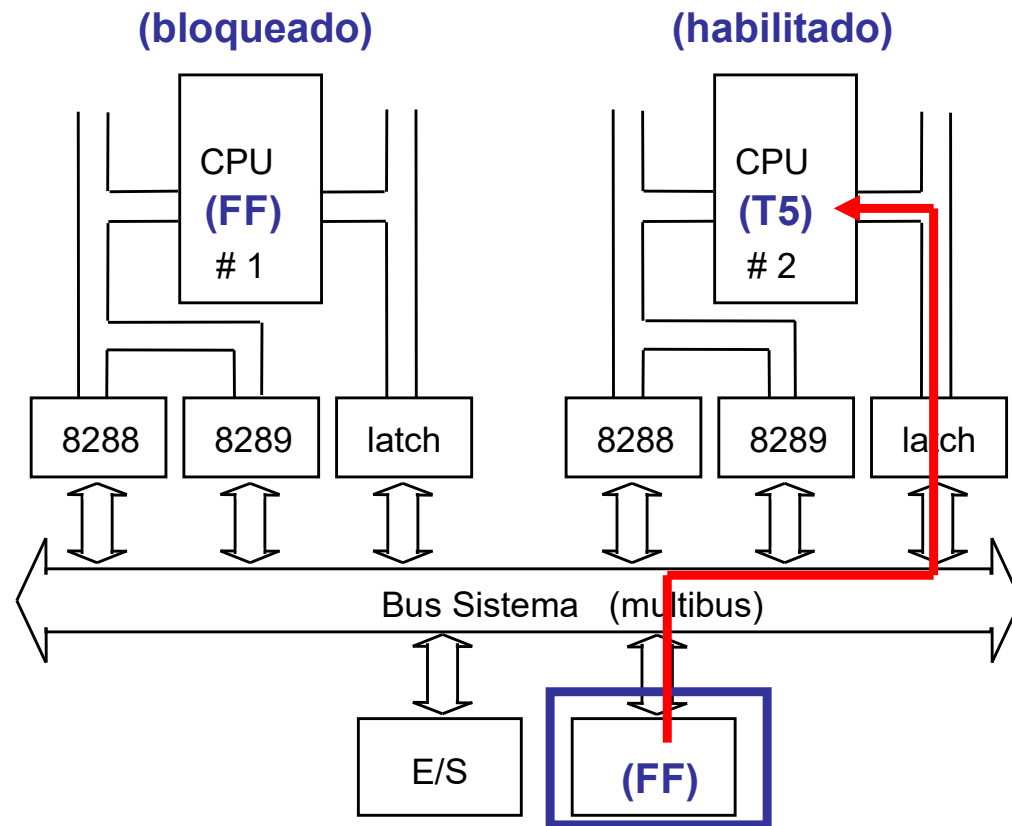
FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

P2

(bloqueada)

P2 lee T4

Multiprocesamiento



Tiempo = 39

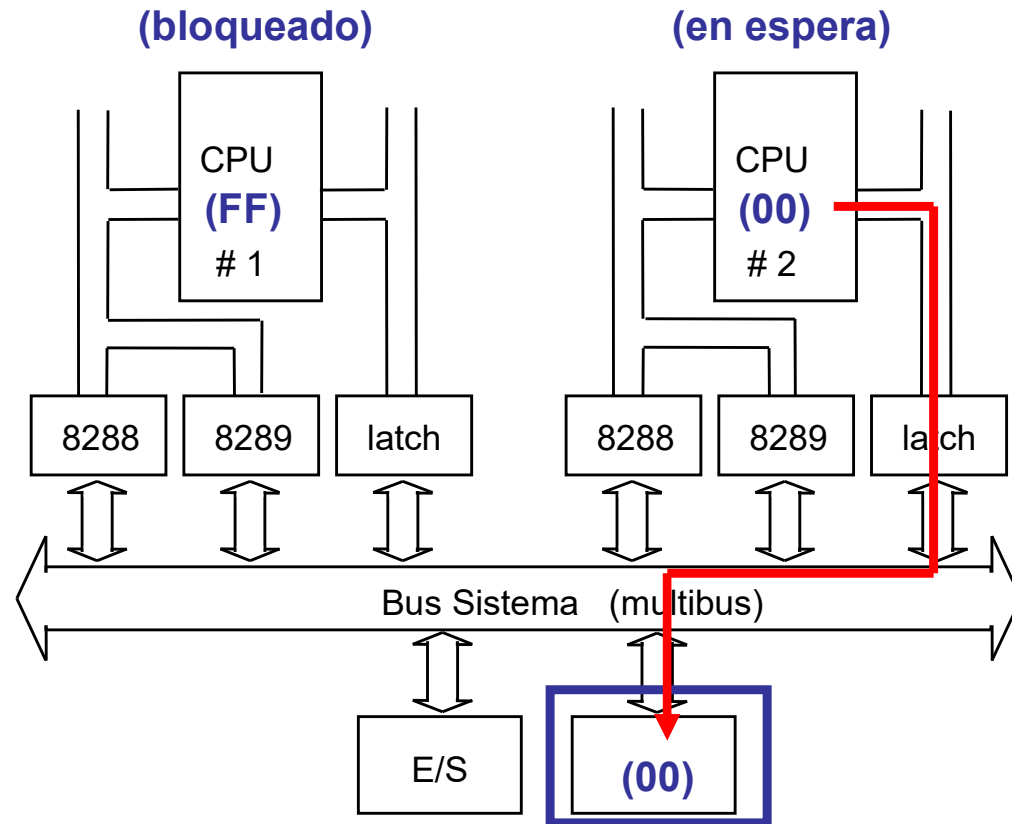
FLAG (FF)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

(bloqueada)

P2

P2 lee T5

Multiprocesamiento



Tiempo = 40

FLAG (00)
T1 (1) ▲
T2 (1) ▲
T3 (1) ▲
T4 (1) ▲
T5 (1) ▲

P2

(habilitada)

P2 escribe 00 en FLAG, liberando la tabla al otro procesador



Fin de la presentación

Técnicas Digitales III – UTN – FRSN

<https://www.frsn.utn.edu.ar/tecnicas3>