

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 399 683**

51 Int. Cl.:

**G06F 11/34** (2006.01)

**G06F 9/52** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **24.11.2008 E 08169785 (6)**

97 Fecha y número de publicación de la concesión europea: **14.11.2012 EP 2192492**

54 Título: **Procedimiento, mecanismo y producto de programa informático para ejecutar varias tareas en un procesador multihilo y para proporcionar estimaciones del peor tiempo de ejecución**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**02.04.2013**

73 Titular/es:

**BARCELONA SUPERCOMPUTING CENTER -  
CENTRO NACIONAL DE SUPERCOMPUTACION  
(100.0%)  
C/JORDÍ GIRONA 31  
08034 BARCELONA, ES**

72 Inventor/es:

**CAZORLA ALMEIDA, FRANCISCO JAVIER**

74 Agente/Representante:

**ZEA CHECA, Bernabé**

**ES 2 399 683 T3**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

## DESCRIPCIÓN

Procedimiento, mecanismo y producto de programa informático para ejecutar varias tareas en un procesador multihilo y para proporcionar estimaciones del peor tiempo de ejecución

[0001] La presente invención se refiere a un procedimiento para ejecutar varias tareas en un procesador multihilo (MT - *multithreaded*), teniendo cada tarea, para cada recurso hardware compartido de un conjunto predeterminado de recursos hardware compartidos en el procesador MT, un valor de tiempo asociado correspondiente a un retraso artificial introducido cuando dicha tarea accede a dicho recurso hardware compartido. Más específicamente, la invención se refiere a un procedimiento para retrasar artificialmente los accesos de hilos que se ejecutan en un procesador multihilo a los recursos hardware compartidos de dicho procesador multihilo, estando configurado por el usuario el retraso artificial que se introduce en cada acceso a los recursos hardware compartidos.

[0002] La invención también se refiere a un mecanismo para ejecutar varias tareas en un procesador MT adecuado para llevar a cabo dicho procedimiento.

[0003] El mecanismo de la invención permite determinar la sensibilidad de una tarea determinada a los conflictos inter-tarea cuando la tarea se ejecuta en un procesador multihilo junto con otras tareas. El mecanismo también permite ejecutar simultáneamente varias tareas en tiempo real estricto (HRT - *Hard Real-Time*) junto con cero o más tareas en tiempo real no estricto (NHRT - *Non Hard-Real Time*) en el mismo procesador MT, de tal manera que se puede proporcionar una estimación segura del peor tiempo de ejecución (WCET - *worst-case execution time*) de las tareas HRT cuando se ejecutan en el procesador MT.

[0004] Preferiblemente, la invención se aplica en sistemas en tiempo real embebidos.

## 20 DEFINICIONES

[0005] Algunos de los términos en toda la siguiente descripción se refieren a componentes de sistema particulares. Como un experto en la técnica apreciará, las compañías de microprocesadores pueden referirse a un componente mediante diferentes nombres. Este documento no tiene la intención de distinguir entre componentes que difieren en el nombre, sino en la función.

[0006] Con el término "acople" o "acoples" o "conectar" se pretende expresar una conexión eléctrica directa o indirecta. Por lo tanto, si un primer dispositivo se acopla a un segundo dispositivo, esta conexión puede ser por medio de una conexión eléctrica directa o por medio de una conexión eléctrica indirecta a través de otros dispositivos y conexiones.

[0007] Con el término "procesador multihilo" se pretende expresar cualquier tipo de procesador que permite ejecutar más de un hilo al mismo tiempo y tiene algunos recursos hardware compartidos entre los hilos. Ejemplos de procesadores multihilo son: procesadores multihilo simultáneo (*simultaneous multithreading processors*), procesadores multihilo de grano fino (*fine grain multithreading processors*), procesadores multihilo de grano grueso (*coarse grain multithreading processors*), procesadores multinúcleo (*multicore processors*), o cualquier combinación de los mismos.

[0008] El término "tarea en tiempo real estricto" ("*Hard Real-Time task*") o "tarea HRT" se refiere a aquellas tareas que requieren una previsibilidad temporal absoluta (*absolute time-predictability*) de tal manera que nunca deberían perderse un límite temporal (*deadline*) de ejecución. En cambio, el término "tarea en tiempo real no estricto" ("*Non-Hard Real-Time task*") o "tarea NHRT" se refiere a las tareas con restricciones de tiempo real débiles o tareas en tiempo real débil (tareas SRT - *soft-real time*), que están más relajadas en el sentido de que ocasionalmente podrían perderse límites temporales; o tareas sin restricciones de tiempo (tareas no en tiempo real o tareas NRT - *Non Real-Time*).

[0009] El término "recurso" se refiere por defecto a un recurso hardware.

[0010] El término "interacción inter-tarea" ("*inter-tarea interaction*") o "conflicto inter-tarea" ("*inter-tarea conflict*") se refiere a la interacción/conflicto que sufre un hilo en los recursos debido a la ejecución de otras tareas. Es decir, es posible decir que una tarea sufre una interacción/conflicto inter-tarea en un recurso compartido cuando dicha tarea va a utilizar un recurso y éste es utilizado por cualquier otra tarea.

## ESTADO DE LA TÉCNICA ANTERIOR

[0011] Tradicionalmente, los procesadores para mercados en tiempo real estricto embebidos (*embedded hard real-time markets*) típicamente se han caracterizado por una arquitectura sencilla, con pipelines cortas y ejecución en orden (*in-order execution*) con el fin de facilitar el cálculo del peor tiempo de ejecución (WCET). Sin embargo, un porcentaje significativo de los sistemas de aplicación en tiempo real requieren o pueden mejorar su funcionalidad, si los procesadores pudieran proporcionar un rendimiento mayor que estas arquitecturas simples. Estas tareas van

desde las aplicaciones en tiempo real débil como aplicaciones de codificación y decodificación de vídeo, hasta aplicaciones en tiempo real estricto como aplicaciones de automoción, aviónicas y sistemas de control industriales.

[0012] Los procesadores multihilo pueden ofrecer este rendimiento adicional requerido por los sistemas en tiempo real. Los procesadores Multihilo (MT) permiten la ejecución de varias tareas al mismo tiempo, lo que conduce tanto a un mayor rendimiento como a una reducción del número de procesadores requeridos en un sistema dado. Por otra parte, la compartición de recursos en estos procesadores les permite tener unos buenos ratios "rendimiento/coste" y "rendimiento/consumo de energía". Esta característica hace que los procesadores MT sean una buena opción para los sistemas en tiempo real embebidos.

[0013] Sin embargo, al mismo tiempo, la compartición de recursos provoca que la ejecución de tareas pase a ser altamente variable, puesto que las tareas interactúan de manera desconocida con las otras tareas en tiempo real estricto (HRT) y en tiempo real no estricto (NHRT). Esto hace que los procesadores MT sean inadecuados para sistemas en tiempo real embebidos. Es decir, hay un gran inconveniente que impide el uso de procesadores MT en los sistemas en tiempo real embebidos: el tiempo de ejecución de las tareas en un procesador MT se vuelve altamente imprevisible dado que las tareas comparten recursos dinámicamente en tiempo de ejecución de forma imprevisible. Bajo estas circunstancias, no se pueden garantizar las restricciones en tiempo real de los sistemas embebidos.

[0014] A nivel de hardware, la imprevisibilidad introducida por la compartición de recursos se ha evitado proporcionando un aislamiento total en la ejecución de tareas en tiempo-real estricto. Es decir, el hardware permite la ejecución de una tarea HRT y tareas NHRT proporcionando una ejecución de la tarea HRT libre de interacciones. En esta categoría se encuentran los documentos [A. El-Haj-Mahmoud, A. S. AL-Zawawi, A. Anantaraman, and E. Rotenberg. *Virtual Multiprocessor: An Analyzable, High-Performance Microarchitecture for Real-Time Computing. Proceedings of the 2005 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES'05)*, pp. 213-224, September 2005] y [Infineon Technologies AG. *TricoreTM2 Architecture Manual*]. En el primer documento, los autores proponen una arquitectura que proporciona una ejecución libre de interferencias entre los hilos en una arquitectura MT que permite la ejecución de hilos HRT. Esta propuesta exige que la arquitectura del procesador sea alterada de manera que no se permite que haya interacción entre hilos. En el último, se permite la ejecución de una tarea NHRT en segundo plano (*background*) sólo cuando la tarea HRT en primer plano (*foreground*) experimenta una ausencia de instrucción en caché (*instruction cache miss*). La tarea NHRT nunca puede retrasar la tarea HRT.

[0015] Con el fin de lograr el aislamiento completo, la arquitectura propuesta impide cualquier interferencia entre los hilos. Esto requiere cambios significativos a nivel de arquitectura: todos los buffers se duplican para cada hilo, las unidades funcionales son asumidas totalmente segmentadas (*pipelined*) por lo que cada una puede procesar una nueva operación por ciclo, las cachés normales no están permitidas, etc.

[0016] En una solicitud de patente europea (del mismo solicitante de la presente solicitud) con el título "Un procesador multihilo y un mecanismo y un procedimiento para ejecutar una tarea en tiempo real estricto en un procesador multihilo" ("*A multithreaded processor and a mechanism and a method for executing one hard real-time task in a multithreaded processor*") se propone un procedimiento y un mecanismo para ejecutar una tarea HRT simple junto con varias tareas NRT y tareas SRT. En la solución propuesta, el software proporciona al hardware el tiempo de holgura (*slack time*) de la tarea HRT. Cada vez que la tarea HRT puede ser retrasada por una tarea NHRT, el hardware extrae este retardo del tiempo de holgura restante. Si el tiempo de holgura restante alcanza un valor mínimo, se detienen las tareas NHRT evitando que la tarea HRT pierda su límite temporal.

[0017] XP 2 538 651 proporciona un estudio de herramientas para medir los peores tiempos de ejecución. El documento XP 2 538 650 describe la introducción de retardos artificiales durante la simulación de una red grande con tiempos considerables de transmisión de paquetes.

[0018] Como se describió anteriormente, la principal fuente de imprevisibilidad en un procesador MT son los conflictos inter-tarea. Es posible decir que una tarea sufre un conflicto inter-tarea en un recurso compartido cuando dicha tarea va a utilizar un recurso y éste es utilizado por cualquier otra tarea. En arquitecturas de hilo único, en las que sólo se ejecuta un hilo a la vez, los hilos casi no sufren interferencias inter-tarea. Sólo cuando una tarea es planificada fuera del procesador y finalmente se vuelve a planificar de nuevo, puede sufrir conflictos inter-tarea, por ejemplo en la caché, ya que la tarea(s) que se estaba ejecutando mientras fue planificada fuera del procesador desalojó parte de sus datos. Sin embargo, dado que las tareas no se ejecutan al mismo tiempo, el número de conflictos inter-tarea es bajo.

[0019] Sin embargo, en un procesador multihilo, los hilos sufren muchos más conflictos inter-hilos, ya que comparten recursos del procesador al mismo tiempo. El efecto que tiene el efecto inter-tarea en una determinada tarea depende de varios factores: el número de recursos compartidos que se están ejecutando en la arquitectura, cuántos hilos comparten dichos recursos al mismo tiempo, y el tipo de recurso.

[0020] Una vez que se ha fijado una determinada arquitectura multihilo, el efecto de los conflictos inter-tarea depende fundamentalmente de las otras tareas con las que una tarea determinada se ha co-planificado. Una metodología común para analizar la sensibilidad de dicha tarea a conflictos inter-tarea consiste en ejecutar dicha tarea en un conjunto de diferentes cargas de trabajo 'estresantes'. Estas cargas de trabajo están compuestas por 5 tareas que estresan los diferentes recursos compartidos, por lo que la tarea en estudio se ve afectada.

[0021] Este procedimiento presenta varias desventajas. En primer lugar, los hilos en las cargas de trabajo estresantes tienen que ser cuidadosamente diseñados para que estresen adecuadamente unos recursos determinados. Por otro lado, es difícil obtener un nivel mínimo de estrés en todos los recursos compartidos al mismo tiempo. Por último, para cada nueva generación de procesadores multihilo en la que cambian los recursos 10 compartidos, es necesario un rediseño de las cargas de trabajo estresantes.

### RESUMEN DE LA INVENCIÓN

[0022] Es un objeto de la presente invención proporcionar un mecanismo para ejecutar varias tareas en un procesador multihilo que simplifique el análisis del comportamiento de dichas tareas cuando sufren conflictos inter-tarea.

15 [0023] La invención se expone en la reivindicación de procedimiento 1 y la reivindicación de sistema 3. Se presenta una reivindicación relativa al correspondiente producto de programa informático, independiente 5.

[0024] Para conseguir lo descrito anteriormente, de acuerdo con un primer aspecto, la invención proporciona un mecanismo adaptado para ejecutar varias tareas en un procesador multihilo (MT), teniendo cada tarea, para cada recurso hardware compartido de un conjunto predeterminado de recursos hardware compartidos en el procesador 20 MT, un retardo temporal artificial asociado que se introduce cuando dicha tarea accede a dicho recurso hardware compartido, comprendiendo el mecanismo medios para establecer, para cada recurso hardware compartido en el conjunto predeterminado de recursos hardware compartidos y cada tarea a retrasar artificialmente, el retraso artificial a introducir en cada acceso de dicha tarea a dicho recurso hardware compartido; medios para verificar si una solicitud está lista para ser enviada a dicho recurso hardware compartido; medios para verificar si el recurso 25 compartido está listo para aceptar dicha solicitud; medios para retrasar dicha petición según el retraso artificial establecido para dicho recurso hardware compartido; medios para permitir que la solicitud de la tarea acceda a dicho recurso hardware compartido.

[0025] El mecanismo de la invención permite determinar la sensibilidad de una tarea determinada a los conflictos inter-tarea cuando se ejecuta en un procesador multihilo. Es posible analizar la sensibilidad de cada tarea a inter-tarea de manera aislada y sin cambios significativos si se añaden de una generación de procesadores a las 30 siguientes algunos cambios en los recursos compartidos.

[0026] De esta manera, el mecanismo retarda artificialmente los accesos de los hilos que se están ejecutando en un procesador multihilo a los recursos hardware compartidos de dicho procesador multihilo. El retraso artificial a introducir en cada acceso a los recursos compartidos es configurado por el usuario.

35 [0027] Básicamente, la arquitectura del procesador multihilo proporciona medios para ejecutar una determinada carga de trabajo,  $workload_{ToStudy}$ , con al menos una tarea en dicho procesador multihilo de tal manera que cuando dicha tarea,  $task_j$ , envía una solicitud para utilizar un recurso compartido,  $r_i$ , y el recurso está listo para ser utilizado, la arquitectura hace la petición de que la tarea  $task_j$  espere a los ciclos del procesador  $ArtificialDelay^r_{taskj}$  antes de que se conceda el acceso a dicho recurso compartido. El procesador multihilo también proporciona medios para 40 permitir que el software configure el retraso a añadir a cada acceso de las tareas en ejecución a los recursos compartidos.

[0028] De acuerdo con un segundo aspecto, la invención proporciona un procedimiento para ejecutar varias tareas en un procesador multihilo (MT), teniendo cada tarea, para cada recurso hardware compartido de un conjunto predeterminado de recursos hardware compartidos en el procesador MT, un retardo de tiempo artificial asociado que 45 se introduce cuando dicha tarea accede a dicho recurso hardware compartido, comprendiendo el procedimiento las etapas de:

a. Establecer, para cada recurso hardware compartido y cada tarea a retrasar artificialmente, el retraso artificial a aplicar a cada acceso de dicha tarea a dicho recurso hardware compartido;

b. Realizar los pasos siguientes (b1) a (b4) en el acceso de cada tarea a retrasar artificialmente a cada recurso 50 hardware compartido del conjunto predeterminado de recursos hardware compartidos en el procesador MT:

b1. Verificar si una solicitud está lista para ser enviada al mencionado recurso hardware compartido;

En caso de resultado positivo,

b2. Verificar si el recurso hardware compartido está listo para aceptar dicha solicitud;

En caso de resultado positivo,

b3. Retrasar dicha solicitud según el retraso artificial establecido para dicho recurso hardware compartido;

b4. Permitir que la solicitud de la tarea acceda a dicho recurso hardware compartido.

5 [0029] De acuerdo con una realización de la invención, el procedimiento puede comprender una etapa (c) de almacenar el retardo artificial establecido en la etapa (a), y una etapa (d) de obtener, antes de la etapa (b3), el retraso artificial almacenado. Dichos retrasos artificiales pueden ser almacenados en una tabla (*DelayTab*) para cada recurso compartido. Esta tabla puede ser implementada, por ejemplo, con un conjunto de registros. La tabla *DelayTab* tiene tantas entradas como el número máximo de tareas que pueden ser retrasadas. En cada entrada, se guarda el retraso artificial a introducir cuando dicha tarea accede a dicho recurso hardware compartido.

10 [0030] Además, la invención proporciona un procesador multihilo (MT) que comprende un mecanismo para ejecutar varias tareas como se ha descrito anteriormente.

15 [0031] De acuerdo con aún otro aspecto, la invención proporciona un programa informático que comprende instrucciones de programa informático para hacer que un ordenador realice el procedimiento para ejecutar varias tareas en un procesador multihilo (MT). Dicho programa informático puede estar embebido en medios de almacenamiento (por ejemplo, en un medio de registro, en una memoria de ordenador o en una memoria de sólo lectura) o portado en una señal portadora (por ejemplo, en una señal portadora eléctrica u óptica).

20 [0032] Por otra parte, la invención también proporciona un mecanismo para ejecutar varias tareas en tiempo real estricto (HRT) de forma simultánea en un multihilo (MT) de tal manera que es posible calcular una estimación WCET segura para las tareas HRT. Dicho mecanismo para ejecutar varias tareas HRT simultáneamente en un procesador multihilo (MT) comprende medios para establecer una serie de parámetros relativos a la configuración del procesador MT; medios para determinar, de acuerdo con el conjunto de parámetros establecido, el retardo máximo, para cada recurso de un conjunto predeterminado de recursos compartidos por al menos dos de las tareas HRT en el procesador MT, que cada una de dichas tareas HRT puede sufrir al acceder a dicho recurso compartido; el  
25 que el retraso artificial establecido para la tarea es el retraso máximo y cada tarea HRT es retrasada cuando accede a cada recurso compartido de acuerdo con dicho retardo artificial; medios para calcular el peor tiempo de ejecución (WCET) para cada tarea HRT, que tiene en cuenta el conjunto de parámetros de configuración del procesador MT; medios para verificar si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT; medios para configurar el procesador MT en un modo en el que no se introduce  
30 retraso artificial alguno a las tareas HRT cuando dichas tareas acceden a los recursos compartidos; medios para ejecutar las tareas HRT según dicho modo.

35 [0033] De acuerdo con una realización de la invención, el mecanismo para ejecutar varias tareas en tiempo real estricto (HRT) simultáneamente en un procesador MT también ejecuta simultáneamente al menos una tarea NHRT, en el que, el conjunto predeterminado de recursos son compartidos por al menos dos tareas de cualquiera de las tareas HRT y la tarea NHRT, en el que, ninguna de las tareas HRT o la tarea NHRT es retrasada artificialmente cuando accede a un recurso compartido, y en el que los medios para ejecutar las tareas HRT ejecutan las tareas HRT y la tarea NHRT.

40 [0034] Básicamente, dicho mecanismo permite el cálculo de un límite superior WCET seguro para las diversas tareas HRT cuando se ejecutan junto con cero o más tareas NHRT en un procesador multihilo con recursos hardware compartidos entre tareas y configurado bajo una configuración  $s$  (es decir, el conjunto de parámetros relativos a la configuración del procesador MT). El mecanismo comprende medios para la vinculación superior (*upper-binding*) del retardo máximo,  $\text{Maxdelay}_s^i$ , que cada tarea HRT puede sufrir debido a la otra interacción con las otras tareas HRT y NHRT al acceder al recurso compartido,  $r_i$ ; medios para ejecutar cada HRT de forma aislada en el procesador multihilo usando el mecanismo que permite determinar la sensibilidad de una tarea dada a los  
45 conflictos inter-tarea cuando se ejecuta en un procesador multihilo, estableciendo la tarea HRT como la  $\text{workload}_{\text{ToStudy}}$ , el  $\text{Maxdelay}_s^i$  calculado para la HRT actual como el  $\text{ArtificialDelay}_{\text{task}_j}^i$ , y medios para proporcionar información de la ejecución de cada tarea HRT a una herramienta externa de estimación WCET. Se asegura que el WCET estimado resultante es mayor que el tiempo de ejecución de dicha HRT cuando se ejecuta simultáneamente con otras tareas NHRT en dicho procesador multihilo bajo la misma configuración  $s$  del procesador. Por  
50 consiguiente, el mecanismo es capaz de ejecutar cargas de trabajo de aplicaciones mezcladas (tareas HRT, SRT y NRT) de manera que el número de procesadores requerido es bajo, reduciendo el coste y el consumo de energía.

55 [0035] Con los mecanismos descritos, puede calcularse el WCET de cada tarea HRT suponiendo que el tiempo para acceder a cada recurso compartido,  $r_i$ , es  $\text{AccessTime}^i$  más  $\text{Maxdelay}_s^i$ , donde  $\text{AccessTime}^i$  es el tiempo máximo real del tiempo de acceso a dicho recurso, si no hay ningún conflicto de recursos inter-tarea, y  $\text{Maxdelay}_s^i$  es el retardo máximo que dicha HRT puede sufrir debido a la interacción con otras tareas HRT y NHRT. En cada configuración  $s$ , el procesador puede tener, por ejemplo, una política de arbitraje diferente entre las tareas cuando

acceden a un recurso compartido. Dependiendo de la política, el valor del tiempo máximo que una tarea HRT tiene que esperar para obtener acceso a un recurso compartido,  $\text{Maxdelay}_s^{r_i}$ , puede cambiar.

[0036]  $\text{Maxdelay}_s^{r_i}$  debe ser establecido de modo que también representa el tiempo más largo en que se retrasa cualquier HRT para obtener acceso al recurso  $r_i$ , en la configuración  $s$  del procesador, debido a la interacción con otras tareas HRT y NHRT.  $\text{Maxdelay}_s^{r_i}$  puede variar dependiendo de la configuración,  $s$ , según la que está configurado el procesador, por ejemplo en un determinado modo  $s1$  el procesador proporciona una política de acceso de retorno al punto de origen (*round robin access policy*) a los recursos compartidos entre las tareas en ejecución, mientras que en un modo  $s2$  el procesador proporciona una política de acceso primero en entrar/primerero en salir (*First In First Out access policy*).

10 [0037] Mediante la adición de un retardo  $\text{Maxdelay}_s^{r_i}$  cuando se accede a un recurso, es posible asegurar que cuando se calcula el WCET se tiene en cuenta la peor situación que la tarea HRT puede abordar cuando se ejecuta con otros hilos. Esto es debido al hecho de que, por la construcción de la arquitectura, se va a conceder el acceso al recurso compartido  $r_i$ , a cualquier HRT en menos de  $\text{Maxdelay}_s^{r_i}$  ciclos después de que dicha tarea HRT haya solicitado acceso al recurso compartido.

15 [0038] Una vez que se ha calculado el WCET de cada tarea HRT y se ha calculado una planificación adecuada, todas las tareas pueden ser ejecutadas de forma segura en el procesador. Cuando las tareas HRT y las tareas NHRT se ejecutan simultáneamente, es decir, el procesador las ejecuta desactivando el primer procedimiento que permite determinar la sensibilidad de una determinada tarea a los conflictos inter-tarea cuando se ejecuta en un procesador multihilo, de modo que el procesador no introduce ningún retraso artificial cuando las tareas acceden a  
20 los recursos compartidos.

[0039] Con el fin de conseguir el objetivo descrito, el mecanismo no requiere modificar el modo en que se calcula el WCET, cambiando el sistema operativo en tiempo real o las propias aplicaciones, ni cambios de hardware significativos (la herramienta de análisis WCET sólo tiene que introducir un determinado retardo,  $\text{Maxdelay}_s^{r_i}$ , cuando se accede a cada recurso compartido). En cuanto a la herramienta WCET, el único cambio observado es considerar  
25 que el acceso de cada recurso es  $\text{AccessTime}^{r_i}$  más  $\text{Maxdelay}_s^{r_i}$  en lugar de sólo  $\text{AccessTime}^{r_i}$ .

[0040] La invención también proporciona un procedimiento para ejecutar varias tareas en tiempo real estricto (HRT) simultáneamente en un procesador multihilo (MT) que comprende las etapas de:

- e. Establecer un conjunto de parámetros relativos a la configuración del procesador MT;
- 30 f. Determinar, de acuerdo con el conjunto de parámetros establecido, el retardo máximo, para cada recurso de un conjunto predeterminado de recursos compartidos por al menos dos de las tareas HRT en el procesador MT, que cada una de dichas tareas HRT puede sufrir cuando accede a dicho recurso compartido;
- g. Ejecutar cada tarea HRT de forma aislada de acuerdo con el procedimiento para ejecutar varias tareas en un procesador MT de acuerdo con cualquiera de las reivindicaciones 1 ó 2, en el que el retraso artificial establecido para la tarea es el retardo máximo determinado en el paso (f) y cada tarea HRT es retrasada al acceder a cada  
35 recurso compartido de acuerdo con dicho retraso artificial;
- h. Calcular el peor tiempo de ejecución (WCET) para cada tarea HRT de acuerdo con la ejecución de dicha tarea HRT en la etapa (g), que tiene en cuenta el conjunto de parámetros de la configuración del procesador MT;
- i. Comprobar si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT;
- 40 En caso de resultado positivo,
- j. Configurar el procesador MT en un modo en el que no se introduce ningún retraso artificial a las tareas HRT cuando dichas tareas acceden a recursos compartidos;
- k. Ejecutar las tareas HRT de acuerdo con dicho modo.

[0041] Dicho procedimiento incluye determinar la holgura (*slack*) disponible para la tarea HRT, y calcular una  
45 planificación factible para las tareas. El procedimiento también incluye inicializar tanto la configuración del procesador como la tabla *DelayTab* en los diferentes recursos hardware compartidos del procesador con el retardo máximo determinado en el paso (f).

[0042] En cada configuración de dicho procesador multihilo, el procesador puede tener, por ejemplo, una política de arbitraje diferente entre las tareas cuando acceden a un recurso hardware, lo que conduce a un  $\text{Maxdelay}_s^{r_i}$   
50 diferente. Además, el procesador comprende la tabla *DelayTab* que tiene tantas entradas como el número máximo de tareas que pueden ser retrasadas. Cada entrada, guarda un retardo determinado por el software para introducirlo en el acceso de cada tarea,  $\text{task}_j$ , a este recurso hardware  $\text{ArtificialDelay}_{\text{task}_j}^s$ .

[0043] De acuerdo con una realización de la invención, el conjunto de parámetros de la configuración del procesador MT comprende al menos uno de los siguientes parámetros: el número de tareas HRT a ejecutar, el número de tareas NHRT a ejecutar, y la política de arbitraje que concede acceso a cada recurso compartido.

[0044] De acuerdo con otra realización de la invención, en caso de resultado negativo en la etapa (i), el control del procedimiento pasa a la etapa (e).

[0045] En otra realización, en la etapa (i), el WCET calculado para cada tarea HRT se ajusta a la planificación adecuada determinada de las tareas HRT cuando el WCET de cada tarea HRT es menor que el límite temporal (*deadline*) de dicha tarea HRT.

[0046] Además, la invención proporciona un procesador multihilo (MT) que comprende un mecanismo para ejecutar las tareas HRT simultáneamente como se describe anteriormente. En cada configuración de dicho procesador multihilo, el procesador puede tener, por ejemplo, una política de arbitraje diferente entre las tareas cuando acceden a un recurso, lo que conduce a un  $\text{Maxdelay}_i^s$  diferente. Además, el procesador puede comprender la tabla DelayTab descrita.

[0047] Finalmente, la invención también proporciona un programa informático que comprende instrucciones de programa para hacer que un ordenador realice el procedimiento para ejecutar varias tareas en un procesador multihilo (MT). Dicho programa informático puede estar embebido en medios de almacenamiento (por ejemplo, en un medio de registro, en una memoria de ordenador o en una memoria de sólo lectura) o portado en una señal portadora (por ejemplo, en una señal portadora eléctrica u óptica).

[0048] Además, la invención proporciona un procedimiento para ejecutar varias tareas en tiempo real estricto (HRT) y al menos una tarea en tiempo real no estricto (NHRT) simultáneamente en un procesador multihilo (MT) que comprende las etapas de:

e. Establecer un conjunto de parámetros relativos a la configuración del procesador MT;

f. Determinar, de acuerdo con el conjunto de parámetros establecido, el retardo máximo, para cada recurso de un conjunto predeterminado de recursos compartidos por al menos dos tareas de cualquiera de las tareas HRT y la tarea NHRT en el procesador MT, que cada una de dichas tareas HRT pueden sufrir al acceder a dicho recurso compartido;

g. Ejecutar cada tarea HRT de forma aislada de acuerdo con el procedimiento para ejecutar varias tareas en un procesador MT de acuerdo con cualquiera de las reivindicaciones 1 o 2, en el que el retraso artificial establecido para la tarea es el retardo máximo determinado en el paso (f) y se retrasa cada tarea HRT al acceder a cada recurso compartido de acuerdo con dicho retraso artificial;

h. Calcular el peor tiempo de ejecución (WCET) para cada tarea HRT de acuerdo con la ejecución de dicha tarea HRT en la etapa (g), que tiene en cuenta el conjunto de parámetros de la configuración del procesador MT;

i. Comprobar si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT;

En caso de resultado positivo,

j. Configurar el procesador MT en un modo en el que no se introduce ningún retraso artificial a las tareas HRT y la tarea NHRT cuando dichas tareas acceden a recursos compartidos;

k. Ejecutar las tareas HRT de acuerdo con dicho modo.

#### BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0049] A continuación se describirá una realización de la presente invención, sólo a modo de ejemplo no limitativo, con referencia a los dibujos adjuntos, en los que:

La figura 1 es una representación esquemática de los parámetros comunes que se utilizan al planificar tareas en tiempo real;

La figura 2 es un diagrama de flujo que ilustra el procedimiento para ejecutar varias tareas HRT junto con cero o más tareas NHRT simultáneamente en un procesador multihilo (MT), de modo que es posible calcular una estimación WCET segura para las tareas HRT;

La figura 3 es un diagrama de flujo que ilustra el procedimiento para ejecutar varias tareas en un procesador MT que introduce un retardo artificial cuando ciertas tareas especificadas acceden a ciertos recursos especificados, siendo usado dicho procedimiento por el procedimiento de la figura 2;

La figura 4 es una representación esquemática del mecanismo para ejecutar varias tareas en un procesador MT que introduce un retardo artificial cuando ciertas tareas especificadas acceden a ciertos recursos especificados, de acuerdo con la invención;

La figura 5 es una representación esquemática del mecanismo para ejecutar varias tareas HRT junto con cero o más tareas NHRT simultáneamente en un procesador multihilo (MT), de modo que es posible calcular una estimación WCET segura para las tareas HRT, de acuerdo con la invención;

La figura 6 es una representación esquemática de un ejemplo de arquitectura de un procesador MT que comprende el mecanismo de la figura 5;

La figura 7 ilustra la temporización de un acceso en la arquitectura mostrada en la figura 5 cuando dicho acceso no sufre ningún conflicto de recursos; y

Fig.8 ilustra la temporización de un acceso en la arquitectura mostrada en la figura 5 cuando dicho acceso sufre un conflicto de recursos.

### DESCRIPCIÓN DE REALIZACIONES

[0050] En la siguiente descripción detallada, se describirá un procesador multihilo (MT) que comprende un mecanismo para ejecutar varias tareas y un mecanismo para ejecutar tareas HRT simultáneamente de acuerdo con la invención. En esta realización preferida, el procesador está configurado de acuerdo con una configuración s.

[0051] Una tarea en tiempo real,  $i$ , generalmente se caracteriza por los parámetros que se muestran en la figura 1:

- Límite temporal ( $d_i$ ): es el tiempo antes del cual la tarea tiene que haberse ejecutado;

- El peor tiempo de ejecución ( $WCET_i = C_i$ ): es un límite superior del tiempo de ejecución de la tarea, y sigue la relación  $WCET_i \leq d_i$ ;

- Tiempo real de ejecución ( $Ret_i^j$ ), para cada instancia,  $j$ , de la tarea: es el tiempo que una instancia de la tarea necesita para ser ejecutada, y sigue la siguiente relación  $Ret_i^j \leq WCET_i$ .  $RET_i$  es el tiempo medio de ejecución de las diferentes instancias de la tarea,

$$RET_i = \frac{1}{N} \cdot \sum_{j=1}^N RET_i^j$$

- Si la tarea es periódica, otro parámetro dado es el período  $P_i$ , que determina el tiempo hasta que la siguiente instancia de la tarea llega al sistema. Por motivos de simplicidad, normalmente en sistemas periódicos, el límite temporal de cada tarea se establece igual a su período, lo que significa que una tarea tiene que ser ejecutada antes de que la siguiente instancia de la tarea llegue al sistema.

- La holgura estática (*static slack*) corresponde a la diferencia entre el WCET de una tarea y su periodo  $P$ ,  $Holgura\text{-}Estática = P - WCET$ . Se denomina estática porque es conocida antes de la ejecución de la aplicación.

- La holgura dinámica (*dynamic slack*) corresponde a la diferencia entre el tiempo medio de ejecución de una tarea  $RET$  y su WCET,  $Holgura\text{Dinámica} = WCET - RET$ .

[0052] En general, cuando se calcula el WCET para una tarea HRT dada, se supone que la tarea tiene acceso completo a todos los recursos del procesador. Sin embargo, este no es el caso para las arquitecturas MT puesto que, en tiempo de ejecución, la tarea HRT compartirá recursos con otros hilos. Nótese que en entornos MT el planificador de tareas debe saber cómo puede afectar la carga de trabajo al WCET. Redefinir el WCET como el mayor tiempo de ejecución en una carga de trabajo arbitraria no es una opción. Seleccionando cuidadosamente una carga de trabajo, podría hacerse que el WCET para una tarea HRT dada sea arbitrariamente grande. Además, las aproximaciones analíticas al WCET fracasarían miserablemente si necesitaran tener en cuenta otro hilo.

[0053] Si la arquitectura del procesador proporciona un aislamiento completo en la ejecución de la tarea HRT, entonces las tareas NHRT sólo tendrían acceso a esos recursos no utilizados por el hilo HRT y la tarea HRT no sería retrasada por las tareas NHRT. Sin embargo, en una arquitectura multihilo estándar, los recursos son compartidos entre hilos en ejecución de manera que los hilos interactúan entre ellos. Proporcionar un aislamiento completo requerirá cambios en el hardware de manera que se evite cualquier interacción entre las tareas HRT. Con el fin de lograr este objetivo se proponen cambios significativos de la arquitectura: se duplican todos los buffers para cada hilo, las unidades funcionales se suponen totalmente segmentadas (*pipelined*) de manera que cada una puede procesar una nueva operación por ciclo, no se permiten cachés normales, etc. Además, la compartición de recursos es el elemento clave de las arquitecturas multihilo para proporcionar un alto rendimiento y unos buenos ratios rendimiento/coste y rendimiento/consumo de energía.

[0054] En contraste con el estado de la técnica descrito, la presente invención permite proporcionar un WCET seguro para tareas cuando se ejecutan en un procesador multihilo con recursos compartidos. El mecanismo de acuerdo con la invención proporciona un límite superior (*upper bound*) del tiempo en que puede retrasarse una tarea al acceder a un recurso compartido determinado,  $Maxdelay^i_s$ . Este límite superior depende del número de tareas HRT y NHRT que se están ejecutando en la configuración del procesador. Una vez que se ha seleccionado una configuración de procesador, puede configurarse el mecanismo de acuerdo con la invención para introducir dicho retardo  $Maxdelay^i_s$  en todos los accesos de una tarea a dicho recurso compartido,  $r_i$ . Cuando se calcula el WCET con una herramienta dada, el mecanismo de acuerdo con la invención añade un retardo de  $Maxdelay^i_s$  a cada acceso a cada recurso compartido durante el cálculo del WCET, haciendo que el WCET proporcionado para las tareas sea un límite superior seguro del tiempo de ejecución de las tareas cuando se están ejecutando simultáneamente en una arquitectura multihilo con recursos compartidos.

[0055] El objeto de la presente realización es que el procesador ejecute varias tareas HRT junto con cero o más tareas NHRT simultáneamente en un procesador multihilo (MT), de modo que es posible calcular una estimación WCET segura para las tareas HRT. El procesador comprende una herramienta para ejecutar dichas tareas, es decir, el mecanismo que permite la ejecución de varias tareas HRT y NHRT simultáneamente en un procesador MT, garantizando que las tareas HRT terminarán antes de su WCET. Dicho mecanismo comprende los medios (hardware y/o software) necesarios para ejecutar el siguiente procedimiento (véase la figura 2).

[0056] En primer lugar, a partir de un estado de reposo, en la etapa (e), el mecanismo establece la configuración s del procesador MT, es decir, establece el conjunto de parámetros relacionados con la configuración del procesador. A continuación, en una etapa (f), de acuerdo con la configuración s establecida, se determina para cada recurso compartido de los recursos compartidos del procesador, el retardo máximo que cada tarea HRT puede sufrir al acceder a dicho recurso compartido (debido a la interacción con la otra tarea HRT y la tarea NHRT).

[0057] A continuación, con el fin de realizar el siguiente paso del procedimiento, el mecanismo, en una etapa (g), utiliza un segundo mecanismo que puede ejecutar varias tareas en un procesador MT introduciendo un retraso artificial cuando ciertas tareas especificadas acceden a ciertos recursos especificados. Esto es, dicho mecanismo permite introducir retrasos artificiales a una tarea HRT cuando dicha tarea accede a un recurso compartido.

[0058] Dicho segundo mecanismo comprende los medios necesarios (hardware/software) para ejecutar el siguiente segundo procedimiento (ver Fig. 3).

[0059] En una etapa (a), el segundo mecanismo establece, para cada recurso hardware compartido y cada tarea a retrasar de forma artificial, el retraso artificial (es decir, el retardo máximo obtenido, para cada tarea y para cada recurso compartido, en el paso (j)) a aplicar a cada acceso de la tarea HRT a dicho recurso compartido. Después, para cada tarea y para cada recurso compartido, el segundo mecanismo verifica, en una etapa (b1), si una petición de la tarea está lista para ser enviada al recurso compartido a acceder. En caso de resultado positivo, en una etapa (b2), el segundo mecanismo verifica si el recurso compartido está listo para aceptar dicha solicitud, de modo que, en caso de resultado positivo, el segundo mecanismo, en una etapa (b3), retrasa la solicitud de la tarea según el retardo máximo (es decir, el retraso artificial) establecido para dicho recurso compartido y, en una etapa (b4), el mecanismo permite que la solicitud de la tarea acceda a dicho recurso compartido.

[0060] Por otro lado, es importante destacar que los retrasos máximos establecidos son almacenados (en una etapa (c)) en el procesador, y más específicamente en una tabla (DelayTab) que puede ser implementada con un conjunto de registros, teniendo dicha tabla tantas entradas como el número máximo de tareas que pueden ser retrasadas. En cada entrada, se guarda el retraso artificial a añadir a dicha tarea. En este caso, el retardo artificial es el retardo máximo que una HRT puede sufrir para conseguir acceso al recurso compartido cuando se ejecuta con la otra tarea HRT y la tarea NHRT, estando el procesador configurado según la configuración s.

[0061] Secuencialmente, cada tarea HRT se ejecuta de forma aislada y se aplican retrasos artificiales de acuerdo con cada recurso compartido, el primer mecanismo sigue la ejecución del primer procedimiento. A continuación, volviendo a la figura 2, en una etapa (h), el primer mecanismo calcula el WCET para cada tarea HRT de acuerdo con el resultado obtenido con la ejecución del primer procedimiento, teniendo en cuenta la configuración s. A continuación, el primer mecanismo, en una etapa (i), verifica si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT, es decir, el primer mecanismo verifica si es posible ejecutar las tareas HRT y la tarea NHRT simultáneamente (el WCET de cada tarea HRT es menor que el límite temporal de dicha tarea HRT). En caso de resultado positivo, en una etapa (j), el primer mecanismo configura el procesador en un modo en el que no se aplica ningún retraso artificial a las tareas cuando dichas tareas acceden a recursos compartidos. En caso de resultado negativo en la etapa (i), el control de dicho primer procedimiento pasa a la etapa (e), es decir, es necesario configurar el procesador de acuerdo con otra configuración. Si no se encuentra ninguna configuración del procesador adecuada, el proceso termina.

[0062] Finalmente, en una etapa (k), el primer mecanismo ejecuta las tareas HRT y la tarea NHRT simultáneamente de acuerdo con el modo configurado en la etapa (j).

[0063] La figura 4 ilustra los componentes del mecanismo para realizar el procedimiento para demorar artificialmente el acceso, de hilos que se están ejecutando en un procesador multihilo, a los recursos hardware compartidos de dicho procesador multihilo. En un procesador multihilo 120 determinado, para cada recurso compartido 130 el segundo mecanismo comprende un Módulo de Unidad de Control de Retraso Artificial (ADCU - *Artificial Delay Control Unit*) 140 que está compuesto por la tabla DelayTab 142 y un multiplexor 144. Una posible implementación del módulo 142 (tabla DelayTab) es por medio de un conjunto de registros hardware. Cada una de las entradas de la DelayTab guarda el retardo a introducir en la ejecución de cada una de las tareas en ejecución. El módulo ADCU 140 comprende además un multiplexor 144 que selecciona de la DelayTab la entrada apropiada en función del número del hilo que está accediendo al recurso compartido que se guarda en el registro 143. Se asume, sin pérdida de generalidad, que el árbitro 135 es capaz de proporcionar el número del hilo que está listo para acceder al recurso compartido. Si no hay un árbitro, otro módulo dentro del recurso compartido 130 proporciona el número de hilo.

[0064] El árbitro 135 concede acceso a las diferentes tareas a ese recurso. El árbitro puede aplicar políticas de acceso diferentes que se pueden configurar desde el software. Conexiones adicionales pueden ser posibles en el interior del árbitro y otros componentes en el recurso compartido, no mostrados en la figura 4.

[0065] El módulo 110 establece la DelayTab de los diferentes recursos compartidos (130 en este caso). La DelayTab de los diferentes recursos se llena secuencialmente utilizando el canal 121. Una vez que se ha llenado la DelayTab en cada recurso, el módulo 115 inicia la ejecución de las tareas. Este módulo es el mismo que en cualquier otra arquitectura multihilo estándar.

[0066] Durante la ejecución de las tareas en cada acceso a un recurso compartido, el multiplexor 144 es activado con el número del hilo que está accediendo al recurso compartido. Una vez que la solicitud y el recurso compartido están listos, la ADCU retrasa el acceso al recurso compartido el tiempo (que puede ser expresado en ciclos de procesador) leído de la tabla DelayTab.

[0067] Según una realización preferida de la invención, dichos módulos 110 (que establecen la DelayTabTable) y 115 (módulo de ejecución) se implementan en forma de un programa informático. Los otros módulos en la figura 4 se implementan en hardware.

[0068] La figura 5 muestra cómo integrar el procedimiento para retrasar artificialmente el acceso de los hilos que se están ejecutando en un procesador multihilo a los recursos hardware compartidos de dicho procesador multihilo, con el procedimiento que permite calcular un peor tiempo de ejecución (WCET) seguro de varias tareas en tiempo real estricto (HRT) bajo una configuración de procesador determinada en la que se ejecutan junto con cero o más tareas en tiempo real débil (SRT) y tareas en tiempo real no estricto (NRT) en un procesador multihilo.

[0069] En la figura 5 el procesador multihilo 300 tiene los medios para implementar el procedimiento para retrasar artificialmente el acceso de los hilos que se ejecutan en un procesador multihilo a los recursos hardware compartidos de dicho procesador multihilo según se describe en la figura 4.

[0070] Inicialmente el módulo 350 llama al módulo 355 el cual establece la configuración del procesador. La configuración del procesador puede incluir, por ejemplo, una política de arbitraje diferente entre las tareas cuando acceden a un recurso compartido. A continuación, la llamada al módulo 350 llama al módulo 365 que se encarga de calcular el retardo máximo que puede sufrir cada acceso de una tarea HRT cuando accede a cada recurso compartido,  $Maxdelay^i_s$ , teniendo en cuenta la configuración actual del procesador. El módulo de cálculo 365 llama al 310 (que corresponde al módulo 110 en la fig. 4) pasándole los valores de  $Maxdelay^i_s$ . El módulo 310 utiliza estos valores como retrasos artificiales a añadir a los accesos de los hilos que se están ejecutando. Por lo tanto, cada vez que la tarea HRT está lista para acceder a dicho recurso compartido, el procesador introduce un retardo de  $Maxdelay^i_s$ , antes de que la tarea HRT pueda utilizar el recurso compartido. Al hacer esto, se está retrasando la tarea HRT el tiempo máximo que se puede retrasar cuando se ejecuta junto con otras tareas HRT y NRT.

[0071] A continuación la llamada al módulo 350 llama al módulo 360 que selecciona una tarea HRT y se la pasa al módulo 315 (que corresponde al módulo 115 en la figura 4), que inicia dicha tarea HRT. Una vez que ha terminado la simulación, el módulo 370 recopila datos acerca de la ejecución de dicha tarea HRT. Una vez que ha acabado el módulo 370, envía la información al módulo 375 que estima un WCET para la HRT. Esto se puede hacer con cualquiera de los procedimientos actuales, tales como análisis estático del WCET (*static WCET analysis*) o análisis basado en medición (*measurement-based analysis*). El WCET resultante, es un límite superior del tiempo de ejecución HRT cuando se ejecuta con otras tareas de acuerdo con la configuración actual del procesador. A continuación, el módulo de estimación 375 envía esta información para llamar al módulo 350, que repite el mismo proceso para todas las tareas HRT.

[0072] Una vez que el módulo 350 tiene una estimación WCET para cada tarea HRT, éste envía esta información sobre el WCET de cada HRT al módulo 380. El módulo 380 intenta construir una planificación factible teniendo en cuenta el límite temporal de las tareas HRT y las otras NHRT a planificar.

[0073] Si se encuentra una planificación factible, el módulo de planificación 380 llama al módulo de cálculo 365 que pone todos los retrasos a cero y se los pasa al módulo 310. El módulo 310 escribe estos valores como retrasos

artificiales en la DelayTab de cada recurso compartido del procesador. En este punto, puede iniciarse la ejecución de la tarea establecida de manera que no se añade un retraso artificial al acceso a los recursos compartidos, y el WCET estimado es un límite superior seguro del tiempo de ejecución de las tareas HRT cuando se ejecutan con otras tareas NHRT compartiendo los recursos del procesador multihilo, bajo la configuración actual del procesador.

- 5 [0074] Si no se encuentra una planificación factible, la llamada al módulo 350 busca otra configuración del procesador que conduzca a una planificación factible. Si no se encuentra la planificación, la tarea establecida no puede ser planificada. En ese caso, es necesario cambiar la tarea establecida.

### Ejemplo

- 10 [0075] La figura 6 muestra una arquitectura multinúcleo (*multicore*) 200 que tiene tres núcleos de cálculo 201, 202 y 203, en la que cada núcleo puede ejecutar un hilo. Todos los núcleos están conectados a una unidad funcional compartida (SFU - *shared functional unit*) 206 por medio de un bus compartido 205.

[0076] El acceso al bus compartido y al bus de la unidad funcional compartida es controlado por un árbitro 220. Cada núcleo se conecta con el árbitro por un canal de control privado (buses) 211, 212 o 213. En dicha arquitectura, se ejecutan dos tareas HRT, HRT1 y HRT2, y una tarea NHRT.

- 15 [0077] Sin pérdida de generalidad y para el propósito de este ejemplo, se supone que el bus y la SFU pueden procesar a la vez sólo una operación de cualquiera de los núcleos. Se supone además que la latencia del bus es de 2 ciclos y la latencia de la SFU es de 5 ciclos. Por lo tanto, la SFU puede aceptar una nueva operación cada 5 ciclos y el bus una cada dos a lo sumo.

- 20 [0078] En su comportamiento normal, cuando una tarea se ejecuta en un núcleo y quiere acceder a la SFU, éste genera una solicitud al árbitro 220. Esta petición se envía al árbitro por uno de los buses de control 211, 212 o 213. Dentro del árbitro, cada núcleo tiene una cola privada de solicitudes 221, 222 o 223, donde se guardan todas las peticiones de ese núcleo hasta que son procesadas. En este ejemplo, se suponen dos entradas por cola de peticiones.

- 25 [0079] En el árbitro 220, una política de selección 235 determina qué solicitudes de todas las solicitudes en las colas de peticiones tienen acceso al bus y a la SFU. El árbitro tiene en cuenta la latencia del bus y de la SFU para evitar que las solicitudes intenten acceder al bus compartido o a la SFU al mismo tiempo.

- 30 [0080] Cuando una tarea no sufre ningún conflicto de recursos con otras tareas, ya sea en el bus o en la SFU, su petición a la SFU sigue la temporización que se muestra en la Fig. 7. En un ciclo dado  $n-1$ , la solicitud se envía al árbitro, por lo que está lista para ser procesada en el siguiente ciclo,  $n$ . En el ciclo  $n$ , el árbitro decide qué petición tiene acceso al bus y la SFU. Dado que sólo hay una petición, se le concede acceso (A). En los ciclos  $n+1$  y  $n+2$  la solicitud utiliza el bus para acceder a la SFU (B). De los ciclos  $n+3$  a  $n+7$  se procesa la operación en la SFU (S).

[0081] Una vez que la operación está lista, se envía su resultado de vuelta al núcleo que la genera. En este caso, se supone que para este propósito hay otro bus y árbitro, en los que se repite el mismo proceso.

- 35 [0082] Cuando varias solicitudes de núcleos diferentes desean acceder a la SFU simultáneamente, el árbitro da prioridad a una solicitud, lo que introduce demoras en las otras solicitudes. Este retardo depende del número de tareas y de la política de selección del árbitro. En este ejemplo se supone que el árbitro da prioridad a la solicitud de las tareas HRT por encima de la solicitud de la tarea NHRT. Además, se supone que las solicitudes de tareas HRT se procesan en forma de retorno al punto de origen (*round robin*).

- 40 [0083] De acuerdo con la invención, es necesario determinar el retardo máximo que una tarea HRT puede sufrir para obtener acceso a un recurso compartido, debido a la interacción con las peticiones de la otra tarea HRT y la tarea NHRT. En este ejemplo,  $\text{MaxDelay}_s^i$  depende del número de tareas HRT, la política de selección del árbitro y el retraso del bus y la SFU. Por ejemplo, supongamos que HRT1 quiere acceder a la SFU en un ciclo determinado  $n-1$ , lo que significa que el árbitro recibe la solicitud en este ciclo y en el ciclo  $n$  podría arbitrar y facilitar el acceso a ese hilo que accedería al bus en el ciclo  $n+1$ . La peor situación que puede surgir es que en el ciclo  $n-1$  se le conceda acceso al bus y a la SFU a una solicitud procedente de la tarea NHRT, y que la tarea HRT2 tenga una solicitud en la cola que tiene prioridad mayor, de acuerdo con la política *round robin*, que la solicitud de HRT1. En esta situación, la temporización seguida por la solicitud de HRT1 se muestra en la figura 8: La solicitud de la tarea NRT es procesada en el ciclo  $n+6$ . En el ciclo  $n+4$ , el árbitro concede acceso a la solicitud de HRT2, de modo que ésta accede al bus en los ciclos  $n+5$  y  $n+6$ , y en el ciclo  $n+7$  estas solicitudes empiezan a ser procesadas. Por lo tanto, la solicitud de HRT1 se retrasa hasta 9 ciclos debido a la interacción con las peticiones de HRT2 y NHRT. Por lo tanto, bajo la configuración del procesador, por ejemplo  $s_0$ , en la que (1) hay dos tareas HRT y una tarea NHRT; y (2) la política de selección del árbitro da prioridad a la solicitud de las tareas HRT sobre las tareas NHRT y utiliza una política *round robin* entre las tareas HRT, el peor retardo que una tarea HRT puede sufrir debido a la interacción con las otras tareas HRT y NHRT,  $\text{MaxDelay}_{s_0}^{\text{bus+SFU}}$ , es de 9 ciclos.

[0084] En este ejemplo particular, se asume una única configuración del procesador, s0, y un bus compartido y unidad de SFU. Una vez que se ha hecho el cálculo del  $\text{Maxdelay}_s^{\text{ri}}$  para el bus y la SFU en este caso, se guarda en la tabla DelayTab de la ADCU asociada a la SFU y bus compartido.

[0085] El siguiente paso, de acuerdo con la invención consiste en calcular el WCET. Esto se puede hacer con cualquier procedimiento existente, por ejemplo cálculo estático del WCET, cálculo del WCET basado en mediciones, etc. Cada HRT se ejecuta de forma aislada. En esta situación, cada vez que una petición de la tarea HRT está lista para ser enviada al bus y la SFU, la ADCU la retrasa en  $\text{Maxdelay}_s^{\text{ri}}$ , que en nuestro ejemplo es igual a 9.

[0086] La información de la ejecución de cada tarea se pasa a un módulo externo (380 en la figura 4.), que la toma en cuenta al calcular una estimación del WCET de cada tarea HRT. Mediante la adición de este retraso en el peor caso,  $\text{Maxdelay}_s^{\text{ri}}$ , en el cálculo del WCET, se tiene en cuenta el retardo máximo que una tarea HRT puede sufrir debido a la interacción con otras tareas.

[0087] Después de calcularse el  $\text{WCET}_s^{\text{ri}}$  para ambas tareas HRT, se reinicia la DelayTab (es decir, se ponen a cero todas las entradas). Luego, un algoritmo de planificación selecciona un algoritmo factible para que se ejecuten las tareas en el procesador multihilo. Se supone que en este ejemplo sólo HRT1, HRT2 y NHRT tienen que ser ejecutadas. El algoritmo de planificación necesita encontrar una configuración del procesador de manera que para cada tarea  $\text{WCET}_s < \text{límite temporal}$ . En el presente ejemplo, se supone que bajo la única configuración del procesador esto es cierto para HRT1 y HRT2 (NHRT no tiene restricciones de tiempo).

[0088] El paso final consiste en ejecutar la carga de trabajo multihilo compuesta por las tareas HRT y la tarea NRT al mismo tiempo. El  $\text{WCET}_s^{\text{ri}}$  calculado para cada tarea HRT es un límite superior seguro del tiempo que les tomará terminar su ejecución.

[0089] Este esquema puede extrapolarse a cualquier arquitectura multihilo en la que hay más recursos compartidos. Por lo tanto, el primer mecanismo de acuerdo con la invención permite la ejecución de una tarea HRT en un procesador multihilo con varias otras tareas NHRT.

#### **Del Retraso Local al Retraso Global**

[0090] La invención también se refiere a un procesador multihilo en el que se ejecutan varias tareas HRT y cero o más tareas NHRT simultáneamente en un procesador multihilo. Si cualquiera de las tareas HRT quiere utilizar un recurso compartido que está siendo utilizado por otra tarea, ésta se retrasa. En nuestra arquitectura propuesta, hay un límite superior del tiempo máximo que la tarea HRT tiene que esperar a la NHRT, MaxDelay. La tarea HRT percibe esta interacción de recursos inter-tarea como un retraso 'extra' en la ejecución de la instrucción que está accediendo a este recurso. Este retardo será referido como retardo local o Ld (Local delay). El retardo local sigue la siguiente regla:  $0 \leq \text{retardo local} \leq \text{MaxDelay}$ .

[0091] En [Thomas Lundqvist and Per Stenström. *Timing Anomalies in Dynamically Scheduled Microprocessors* Thomas. *Proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS)*. pp 12-21. December 1999] y [Jan Reineke, Björn Wachter, Stephan Thesing, Reinhard Wilhelm, Ilia Polian, Jochen Eisinger, Bernd Becker. *A Definition and Classification of Timing Anomalies*. 7th International Workshop on Worst-Case Execution Time (WCET) Analysis. July 2006], se muestra que, dependiendo de la arquitectura del procesador y el modelo de ejecución subyacente (ejecución en orden, ejecución fuera de orden, ejecución superescalar, etc.), un determinado retardo local (Ld - Local delay) en una instrucción de un programa puede resultar en un retardo global en la ejecución del programa, Gd (Global delay), donde  $Gd > Ld$ . O peor aún, el retardo local puede resultar en un retraso global sin límites (*unbound global delay*). Algunas de las, así llamadas, anomalías de tiempo, se producen en procesadores que ejecutan las instrucciones fuera de orden, otras en procesadores con cachés y otras con procesadores que soportan especulación.

[0092] En el marco de la presente invención, no se ha hecho ninguna suposición sobre el modelo de ejecución o la arquitectura del procesador. De hecho, la invención que aquí se presenta se puede aplicar a cualquier arquitectura multihilo en la que un retardo local en el tipo de instrucción que accede a los recursos compartidos resulta en un retraso global limitado (*bound global delay*) o en la que pueden aplicarse las soluciones propuestas en [Thomas Lundqvist and Per Stenström. *Timing Anomalies in Dynamically Scheduled Microprocessors* Thomas. *Proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS)*. pp 12-21. December 1999] y [Alexandru Andrei, Petru Eles, Zebo Peng and Jakob Rosen. *Predictable Implementation of Real-Time Applications on Multiprocessor Systems-on-Chip*. *Proceedings of the 21 st International Conference on VLSI Design*. pp 103-110. 2008] para limitar (*bind*) el retardo global.

[0093] En la discusión en las secciones anteriores, se supone, sin pérdida de generalidad, que el retraso global es igual al retardo local,  $Gd \leq Ld$ .

[0094] Aunque la presente invención ha sido descrita en detalle para fines de ilustración, se entiende que dicho detalle es solamente para ese propósito, y que los expertos en la técnica pueden hacer variaciones en ella sin apartarse del alcance de la invención.

[0095] Así, mientras que las realizaciones preferidas de los procedimientos y de los sistemas se han descrito en referencia al entorno en el que se desarrollaron, son meramente ilustrativas de los principios de la invención. Pueden concebirse otras realizaciones y configuraciones sin apartarse del alcance de las reivindicaciones adjuntas.

[0096] Además, aunque las realizaciones de la invención descritas con referencia a los dibujos comprenden aparatos informáticos y procesos realizados en aparatos informáticos, la invención también se extiende a programas informáticos, particularmente programas informáticos sobre o en un portador, adaptados para poner en práctica la invención. El programa puede estar en forma de código fuente, código intermedio entre código fuente y código objeto tal como en forma parcialmente compilada, o en cualquier otra forma adecuada para su uso en la implementación de los procedimientos según la invención. El portador puede ser cualquier entidad o dispositivo capaz de portar el programa.

[0097] Por ejemplo, el portador puede comprender un medio de almacenamiento, tal como una ROM, por ejemplo un CD ROM o una ROM de semiconductores, o un medio de grabación magnético, por ejemplo un disquete o disco duro. Además, el portador puede ser un portador transmisible tal como una señal eléctrica u óptica, que puede transmitirse a través de cable eléctrico u óptico o por medio de radio u otros medios.

15 [0098] Cuando el programa está embebido en una señal que puede ser transportada directamente por un cable u otro dispositivo o medio, el portador puede estar constituido por dicho cable u otro dispositivo o medio.

[0099] Alternativamente, el portador puede ser un circuito integrado en el que está embebido el programa, estando el circuito integrado adaptado para realizar, o para uso en la realización de, los procesos relevantes.

## REIVINDICACIONES

1. Procedimiento para ejecutar varias tareas en un procesador multihilo, MT, teniendo cada tarea, para cada recurso hardware compartido de un conjunto predeterminado de recursos hardware compartidos en el procesador MT, un retardo temporal artificial asociado que se introduce cuando dichas tareas acceden a dicho recurso hardware compartido, comprendiendo el procedimiento las etapas de:
- 5
- a. Establecer, para cada recurso hardware compartido y cada tarea a retrasar de forma artificial, el retraso artificial a aplicar a cada acceso de dicha tarea a dicho recurso hardware compartido;
  - b. Realizar los siguientes pasos (b1) a (b4) en el acceso de cada tarea a retrasar de forma artificial a cada recurso hardware compartido en el conjunto predeterminado de recursos hardware compartidos en el procesador MT:
- 10
- b1. Verificar si una solicitud está lista para ser enviada a dicho recurso hardware compartido;  
En caso de resultado positivo,
  - b2. Verificar si el recurso hardware compartido está listo para aceptar dicha solicitud;  
En caso de resultado positivo,
  - b3. Retrasar dicha solicitud según el retraso artificial establecido para dicho recurso hardware compartido;
- 15
- b4. Permitir que la solicitud de la tarea acceda a dicho recurso hardware compartido.
2. Procedimiento de acuerdo con la reivindicación 1, que comprende además las etapas de:
- c. Almacenar el retraso artificial establecido en la etapa (a);
  - d. Obtener, antes de la etapa (e), el retraso artificial almacenado.
3. Mecanismo adaptado para ejecutar varias tareas en un procesador multihilo; MT; de acuerdo con el procedimiento de la reivindicación 1, teniendo cada tarea, para cada recurso hardware compartido de un conjunto predeterminado de recursos hardware compartidos en el procesador MT, un retardo temporal artificial asociado que se introduce cuando dicha tarea accede a dicho recurso hardware compartido, en el que el mecanismo comprende medios para establecer, para cada recurso hardware compartido en el conjunto predeterminado de recursos hardware compartidos y cada tarea a retrasar de forma artificial, el retraso artificial a aplicar a cada acceso de dicha tarea a dicho recurso hardware compartido;
- 20
- 25
- medios para verificar si una solicitud está lista para ser enviada a dicho recurso compartido;
- medios para verificar si el recurso compartido está listo para aceptar dicha solicitud;
- medios para retrasar dicha petición según el retraso artificial establecido para dicho recurso compartido; y
- medios para permitir que la solicitud de la tarea acceda a dicho recurso compartido.
- 30
4. Procesador multihilo (MT) **caracterizado por el hecho de que** comprende un mecanismo para ejecutar varias tareas de acuerdo con la reivindicación 3.
5. Producto de programa informático que comprende instrucciones de programa para hacer que un ordenador realice el procedimiento para ejecutar varias tareas en un procesador MT según una cualquiera de las reivindicaciones 1 ó 2.
- 35
6. Producto de programa informático según la reivindicación 5, incorporado en medios de almacenamiento.
7. Producto de programa informático según la reivindicación 5, portado en una señal portadora.
8. Procedimiento para ejecutar varias tareas en tiempo real estricto (HRT) simultáneamente en un procesador multihilo (MT) que comprende las etapas de:
- e. Establecer un conjunto de parámetros relacionados con la configuración del procesador MT;
- 40
- f. Determinar, de acuerdo con el conjunto de parámetros establecido, el retardo máximo, para cada recurso de un conjunto predeterminado de recursos compartidos por al menos dos de las tareas HRT en el procesador MT, que cada una de dichas tareas HRT puede sufrir cuando accede a dicho recurso compartido;
  - g. Ejecutar cada tarea HRT de forma aislada de acuerdo con el procedimiento para ejecutar varias tareas en un procesador MT de acuerdo con cualquiera de las reivindicaciones 1 o 2, en el que el retraso artificial establecido

para la tarea es el retardo máximo determinado en el paso (f) y se retrasa cada tarea HRT al acceder a cada recurso compartido de acuerdo con dicho retraso artificial;

h. Calcular el peor tiempo de ejecución (WCET) para cada tarea HRT de acuerdo con la ejecución de dicha tarea HRT en la etapa (g), que tiene en cuenta el conjunto de parámetros de la configuración del procesador MT;

- 5 i. verificar si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT;

En caso de resultado positivo,

j. Configurar el procesador MT en un modo en el que no se introduce retraso artificial a las tareas HRT cuando dichas tareas acceden a recursos compartidos;

- 10 k. Ejecutar las tareas HRT de acuerdo con dicho modo.

9. Procedimiento para ejecutar varias tareas en tiempo real estricto (HRT) simultáneamente en un procesador MT según la reivindicación 8, y ejecutar simultáneamente al menos una tarea NHRT, en el que, en la etapa (f), el conjunto predeterminado de recursos son compartidos por al menos dos tareas de cualquiera de las tareas HRT y la tarea NHRT, en el que, en la etapa (j), ninguna de las tareas HRT o la tarea NHRT es retrasada de forma artificial al acceder a un recurso compartido, y en el que, en la etapa (k), se ejecutan las tareas HRT y la tarea NHRT.

10. Procedimiento según una cualquiera de las reivindicaciones 8 ó 9, en el que el conjunto de parámetros de la configuración del procesador MT comprende al menos uno de los siguientes parámetros: el número de tareas HRT a ejecutar, el número de tareas NHRT a ejecutar, y la política de arbitraje que concede acceso a cada recurso compartido.

- 20 11. Procedimiento según una cualquiera de las reivindicaciones 8 a 10, en el que, en caso de resultado negativo en la etapa (i), el control del procedimiento pasa a la etapa (e).

12. Procedimiento según una cualquiera de las reivindicaciones 8 a 11, en el que, en la etapa (i), el WCET calculado para cada tarea HRT se ajusta a la planificación determinada adecuada de las tareas HRT cuando el WCET de cada tarea HRT es menor que el límite temporal de dicha tarea HRT.

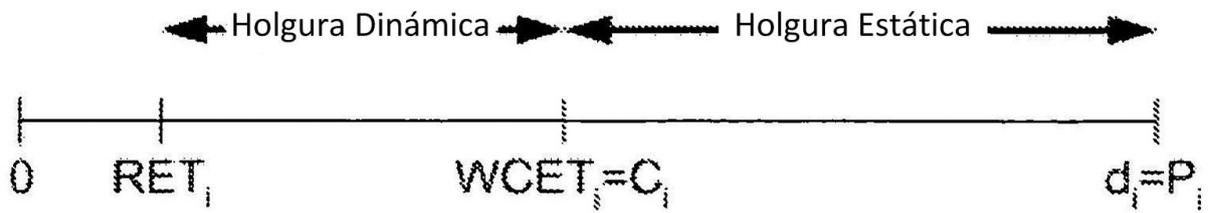
- 25 13. Mecanismo adaptado para ejecutar varias tareas HRT simultáneamente en un procesador multihilo (MT) **caracterizado por el hecho de que** comprende medios para establecer un conjunto de parámetros relativos a la configuración del procesador MT; medios para determinar, de acuerdo con el conjunto de parámetros establecido, el retardo máximo, para cada recurso de un conjunto predeterminado de recursos compartidos por al menos dos de las tareas HRT en el procesador MT, que cada una de dichas tareas HRT pueden sufrir cuando accede a dicho recurso compartido; el mecanismo según la reivindicación 3 para ejecutar varias tareas en un procesador MT para ejecutar cada tarea HRT de forma aislada, en el que el retraso artificial establecido para la tarea es el retraso máximo y se retrasa cada tarea HRT cuando accede a cada recurso compartido de acuerdo con dicho retardo artificial; medios para calcular el peor tiempo de ejecución (WCET) para cada tarea HRT, que tiene en cuenta el conjunto de parámetros de la configuración del procesador MT; medios para verificar si el WCET calculado para cada tarea HRT se ajusta a una determinada planificación adecuada de las tareas HRT; medios para configurar el procesador MT en un modo en el cual no se introduce retraso artificial a las tareas HRT cuando dichas tareas acceden a recursos compartidos; medios para ejecutar las tareas HRT de acuerdo con dicho modo.

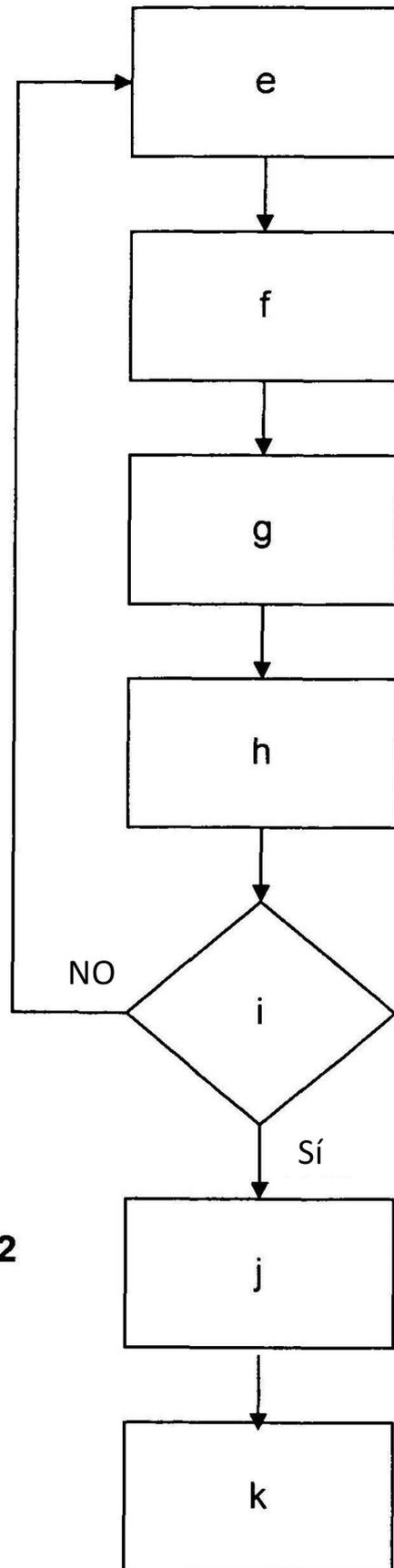
- 40 14. Mecanismo para ejecutar varias tareas en tiempo real estricto (HRT) simultáneamente en un procesador MT según la reivindicación 13, y ejecutar simultáneamente al menos una tarea NHRT, en el que, el conjunto predeterminado de recursos son compartidos por al menos dos tareas de cualquiera de las tareas HRT y la tarea NHRT, en el que, ninguna de las tareas HRT o la tarea NHRT es retrasada artificialmente cuando accede a un recurso compartido, y en el que los medios para ejecutar las tareas HRT ejecuta las tareas HRT y la tarea NHRT.

15. Procesador multihilo (MT) **caracterizado por el hecho de que** comprende un mecanismo para ejecutar varias tareas de acuerdo con una cualquiera de las reivindicaciones 13 ó 14.

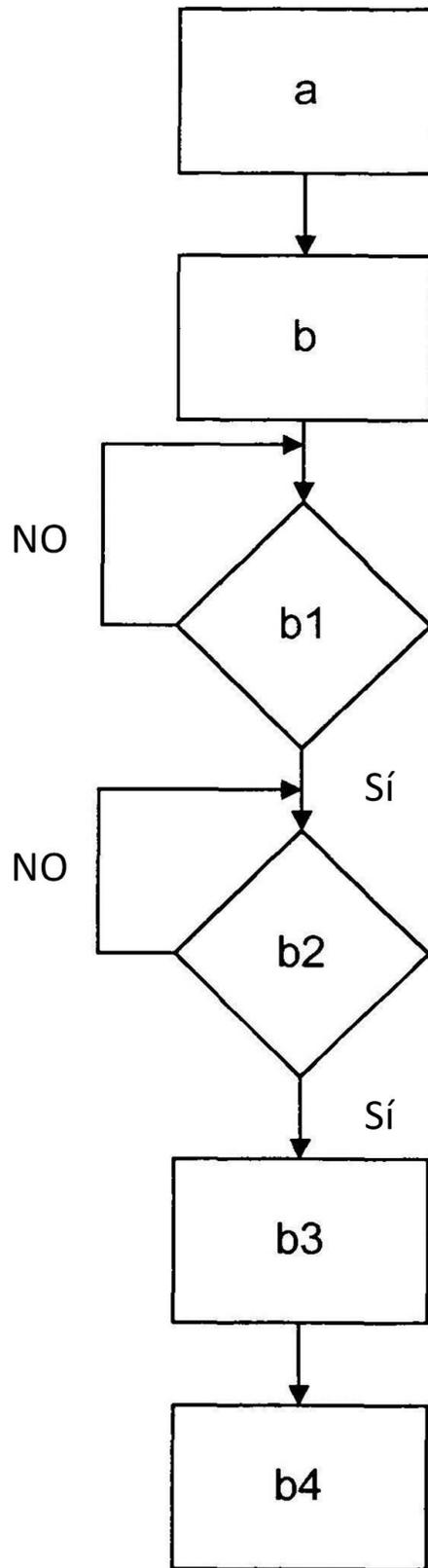
- 45 16. Producto de programa informático que comprende instrucciones de programa para hacer que un ordenador realice el procedimiento para ejecutar varias tareas en un procesador MT según una cualquiera de las reivindicaciones 8 a 12.

**FIG.1**





**FIG.2**



**FIG.3**

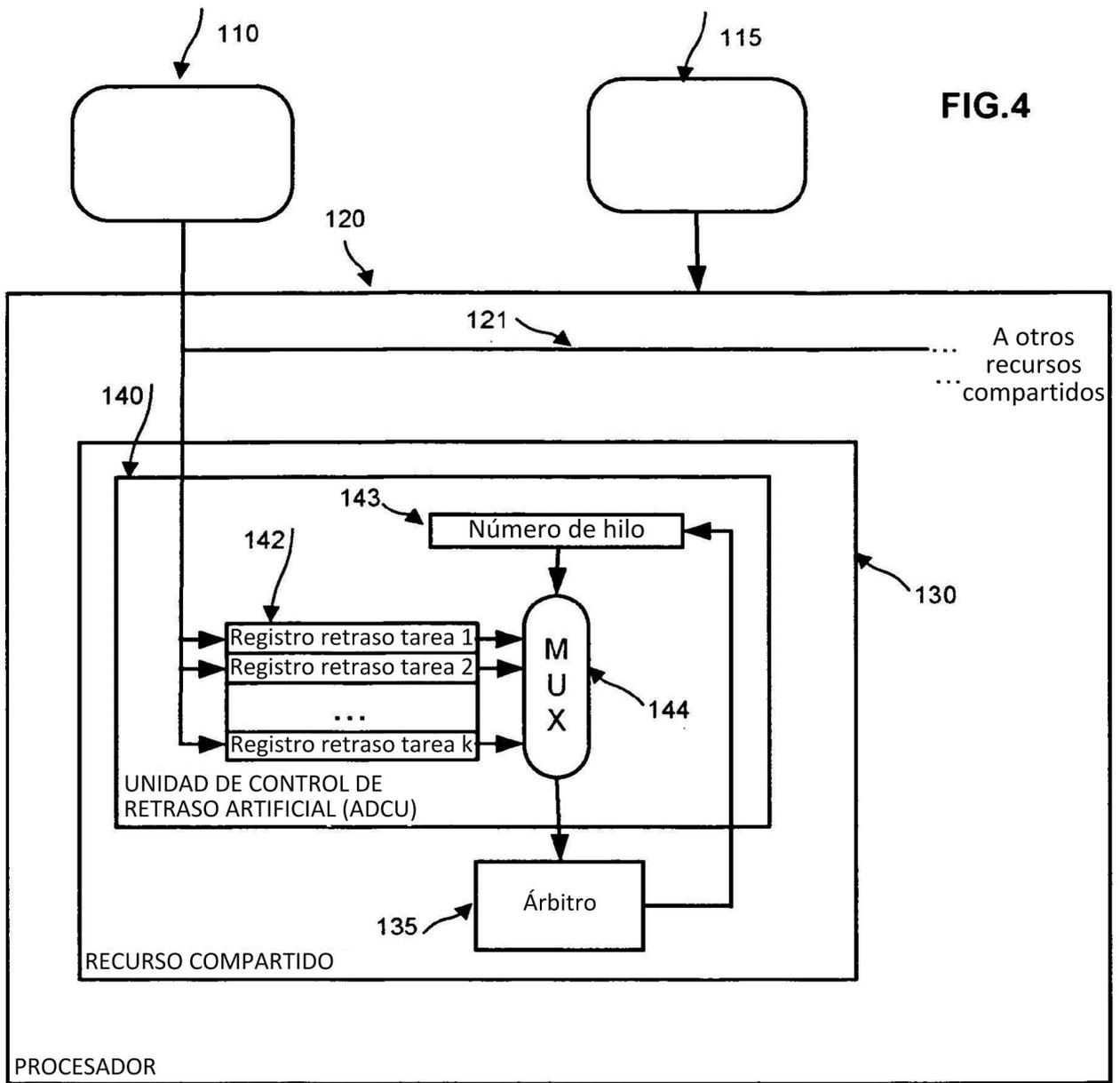
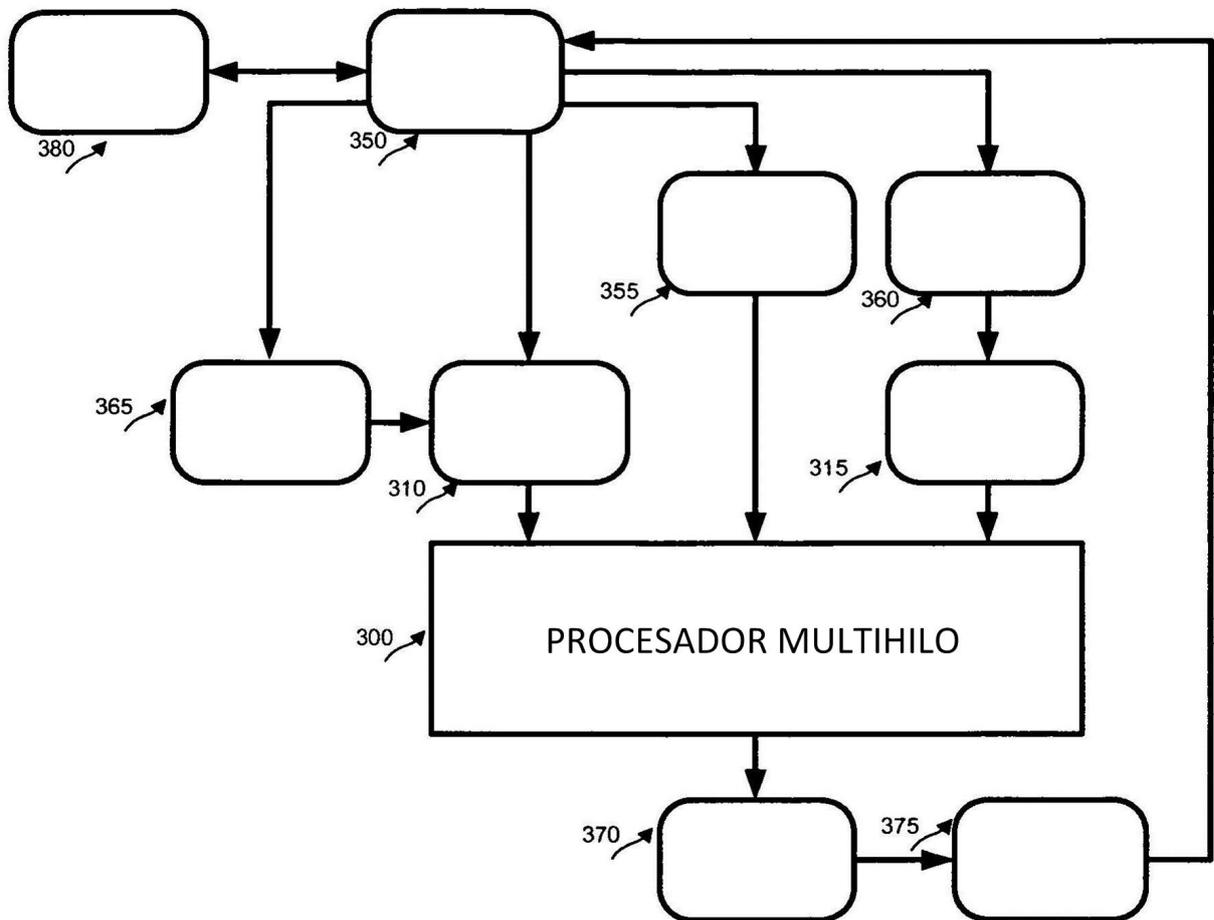


FIG.5



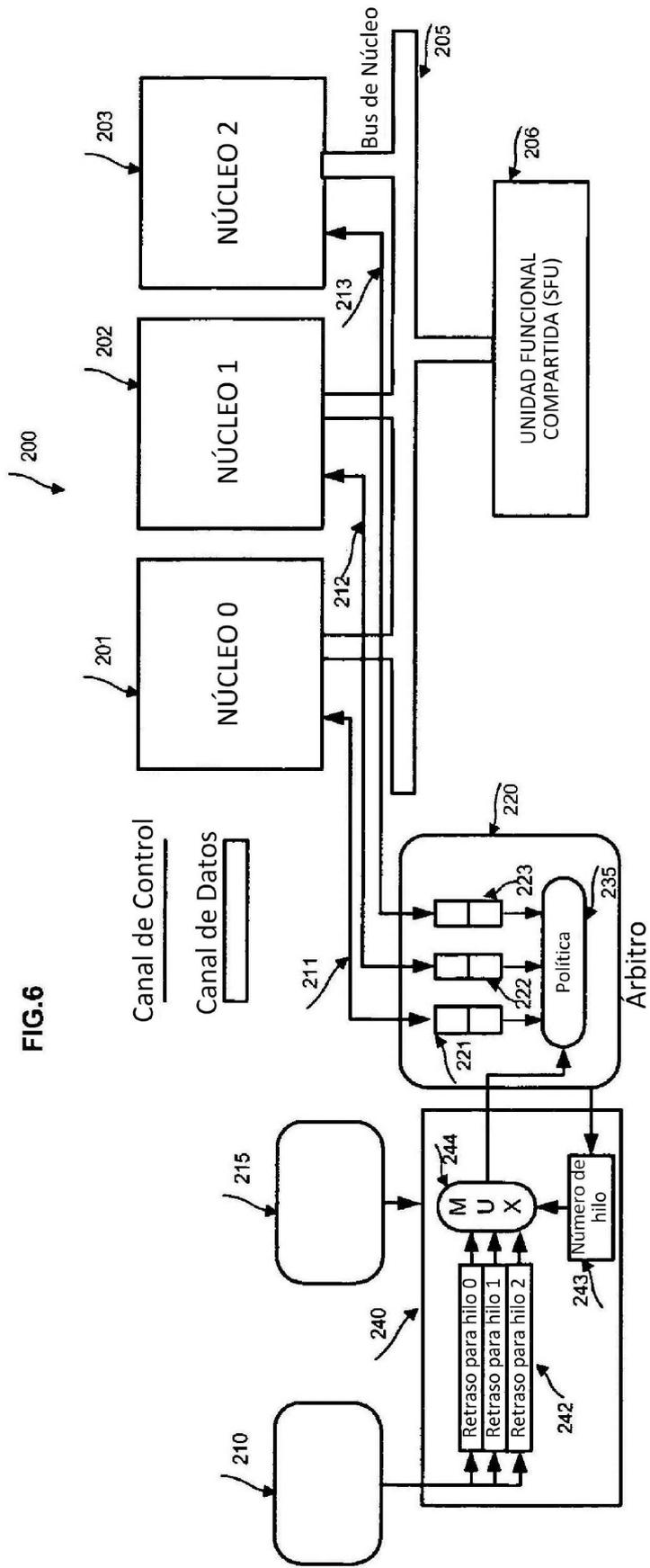


FIG.6

**FIG.7**

	n-1	n+0	n+1	n+2	n+3	n+4	n+5	n+6	n+7
tarea		A	B	B	S	S	S	S	S

	n-1	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7
NHRT	A	B	B	S	S	S	S	S	
HRT1		-	-	-	-	A	B	B	S
HRT2						-	-	-	-

	n+8	n+9	n+10	n+11	n+12	n+13	n+14	n+15	n+16
NHRT									
HRT1	S	S	S	S					
HRT2	-	A	B	B	S	S	S	S	S

**FIG.8**

## REFERENCIAS CITADAS EN LA DESCRIPCIÓN

Esta lista de referencias citadas por el solicitante es únicamente para la comodidad del lector. No forma parte del documento de la patente europea. A pesar del cuidado tenido en la recopilación de las referencias, no se pueden 5 excluir errores u omisiones y la EPO niega toda responsabilidad en este sentido.

## Literatura diferente de patentes citadas en la descripción

- **A. EL-HAJ-MAHMOUD; A. S. AL-ZAWAWI ; A. ANANTARAMAN ; E. ROTENBERG.** Virtual Multiprocessor: An Analyzable, High-Performance Microarchitecture for Real-Time Computing. *Proceedings of the 2005 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES'05, September 2005, 213-224* [0014]
- **JAN REINEKE ; BJÖRN WACHTER ; STEPHAN THESING ; REINHARD WILHELM ; ILIA POLIAN ; JOCHEN EISINGER ; BERND BECKER.** A Definition and Classification of Timing Anomalies. *7th International Workshop on Worst-Case Execution Time (WCET) Analysis, July 2006* [0091]
- **THOMAS LUNDQVIST ; PER STENSTRÖM.** Timing Anomalies in Dynamically Scheduled Microprocessors Thomas. *Proceedings of the 20th IEEE Real- Time Systems Symposium (RTSS, December 1999, 12-21* [0091] [0092]
- **ALEXANDRU ANDREI ; PETRU ELES ; ZEBO PENG ; JAKOB ROSEN.** Predictable Implementation of Real-Time Applications on Multiprocessor Systems-on-Chip. *Proceedings of the 21 st International Conference on VLSI Design, 2008, 103-110* [0092]