

*DIFU*_{100ci@}, Revista de Difusión Científica

Ingeniería y Tecnologías

Vol: 16

Num: 1

Publicación: Enero - Abril 2022

ISSN: 2007-3585



Una publicación de la Universidad Autónoma de Zacatecas "Francisco García Salinas"

Directorio

Universidad Autónoma de Zacatecas “Francisco García Salinas”

Dr. Rubén de Jesús Ibarra Reyes, *Rector*

Dr. Ángel Román Gutiérrez, *Secretario General*

M.C. Hans Hiram Pacheco García, *Secretario Académico*

Dr. Carlos Francisco Bautista Capetillo, *Coord. Investigación y Posgrado*

Editor en Jefe

Víktor Iván Rodríguez Abdalá, *Universidad Autónoma de Zacatecas, (México)*

Editores Asociados

Manuel Hernández Calviño, *Universidad de La Habana, (Cuba)*

José Ricardo Gómez Rodríguez, *Universidad Autónoma de Zacatecas, (México)*

Grupo Revisor

Jorge Flores Troncoso, *Universidad Autónoma de Zacatecas, (México)*

Remberto Sandoval Aréchiga, *Universidad Autónoma de Zacatecas, (México)*

Salvador Ibarra Delgado, *Universidad Autónoma de Zacatecas, (México)*

Juvenal Villanueva Maldonado, *CONACyT-UAZ, (México)*

Jorge Simón Rodríguez, *CONACyT-UAZ, (México)*

Rodrigo Daniel Méndez Ramírez, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Leonardo Acho Zuppa, *Universidad Politécnica de Catalunya, (España)*

Miguel Andrés, *Universidad de Valencia, (España)*

José Ángel González Fraga, *Universidad Autónoma de Baja California, (México)*

Ramón Parra Michel, *Centro de Investigación y de Estudios Avanzados del IPN, (México)*

Leonel Soriano Equigua, *Universidad de Colima, (México)*

José Luis Álvarez Flores, *Universidad de Colima, (México)*

Fermín Marcelo Rubén Maciel Barbosa, *Universidad de Colima, (México)*

Francisco Rubén Castillo Soria, *Universidad Autónoma de San Luis Potosí, (México)*

Cristian Eduardo Boyain y Goytia Luna, *Universidad Autónoma de Zacatecas, (México)*

Juan Pablo Morales Álvarez, *Instituto Tecnológico de La Paz, (México)*

Daniel Armando Hirles Valles, *Instituto Tecnológico de La Paz, (México)*

Jaime Sánchez García, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Salvador Villarreal Reyes, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Ernesto García Domínguez, *Universidad Autónoma de Zacatecas, (México)*

Rafael Villela Varela, *Universidad Autónoma de Zacatecas, (México)*

Jorge de la Torre y Ramos, *Universidad Autónoma de Zacatecas, (México)*

Manuel Reta Hernández, *Universidad Autónoma de Zacatecas, (México)*

Derechos reservados

Editorial

Universidad Autónoma de Zacatecas
“Francisco García Salinas”

Diseño de revista

Víktor Iván Rodríguez Abdalá
José Ricardo Gómez Rodríguez

Derechos Reservados

Universidad Autónoma de Zacatecas
“Francisco García Salinas”

ISSN 2007-3585

DIFU100ci@ (léase difucencia) es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, “Francisco García Salinas”, a través de la Unidad Académica de Ingeniería Eléctrica. Jardín Juárez No. 147, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Tel. (01 492) 925 6690.

Reservas de Derechos al Uso Exclusivo No. 04-2015-060212215100-203, ISSN: 2007-3585, ambos otorgados por el Instituto Nacional de Derecho de Autor.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Universidad Autónoma de Zacatecas “Francisco García Salinas” a través de la Unidad Académica de Ingeniería Eléctrica.

<http://www.uaz.edu.mx>

<http://difu100cia.uaz.edu.mx>

difu100cia@uaz.edu.mx

Portada: <https://www.microchip.com>

Plantilla de revista: ElegantLaTeX

Plantilla de artículos: Wenkeker Article

Carta del editor

Hoy en día, el diseño de hardware se realiza de una manera completamente diferente a como se realizaba en sus inicios, pasando a un diseño asistido por computadora sin necesidad de realizar diseño de circuitos impresos o soldar alguna pieza, esto sin dejar de ser un desarrollo de hardware, ya que se utilizan dispositivos electrónicos programables.

Esto ha permitido que los tiempos de diseño, pruebas y resultados sean menores en comparación a sus inicios, incluso permitiendo el rediseño del hardware sin tocar algún componente electrónico físicamente, sino que esto sea a través de un lenguaje descriptor de hardware. Cabe mencionar que aunque se pudiera pensar que al momento de utilizar un lenguaje esto se puede interpretar como un problema que se puede resolver a través de software, en realidad esto no deja de ser un problema de hardware.

También, gracias a esto, el diseño de hardware electrónico se puede realizar de una forma mas simple con la ayuda de tarjetas de desarrollo, limitando así los problemas que se puedan derivar por la cadena de suministro, esto es, la disponibilidad de ciertos componentes electrónicos en el mercado, ya que estos se podrían diseñar directamente en la tarjeta de desarrollo y probar su funcionamiento en el sistema que se esté diseñando, en caso de que no cumpla con los requisitos que se busca, simplemente se elimina del hardware.

Este tipo de herramientas han permitido que el desarrollo de hardware se suba al ritmo de desarrollo de la actualidad, donde los requerimientos de nuevas tecnologías con un alto grado de desempeño y muy bajo costo son requisitos esenciales, como en el caso del Internet de las Cosas y la Industria 4.0.



Víctor Iván Rodríguez Abdalá
Editor en Jefe

Índice general



Directorio	I
Derechos reservados	II
Carta del editor	III
1. Arquitectura de un regulador de ancho de banda en el sistema de interconexión de un SoC	1

A bandwidth-regulating architecture in the interconnection system of a SoC

Arquitectura de un regulador de ancho de banda en el sistema de interconexión de un SoC

Salvador Ibarra-Delgado^{*1}, Manuel Hernandez-Calvino¹, Víktor I. Rodríguez-Abdalá¹, J. Ricardo Gómez-Rodríguez¹, Remberto Sandoval-Aréchiga¹, and Jorge Flores-Troncoso¹

¹Universidad Autónoma de Zacatecas (UAZ), Unidad Académica de Ingeniería Eléctrica,
Posgrado en Ingeniería para la Innovación Tecnológica,
Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.
{sibarra,mhcalvinno,abdala,rgrodri,rsandoval,jflorester}@uaz.edu.mx

Abstract

Nowadays, within a chip, it is possible to integrate Processing Elements that collaborate in executing a task to meet the established Quality of Service requirements (latency and throughput). They often need to access shared resources with different regularity to achieve this. In this paper, we show the hardware architecture of a bandwidth regulator for a bus-type Interconnection System (IS) that allows its use in proportion to the support weights of a Round-Robin arbiter. The proportion is achieved by modifying, at run-time, the weights of the arbiter. The regulator's behavior is analyzed by modeling the system in SystemC with clock-cycle precision. The contributions of this paper are:

- Design and hardware modeling of a bandwidth regulator for a bus interconnection system.
- Evaluation of the capacity of the regulator to allocate bandwidth according to the established ratios.
- Determination of the latency derived from the use of the regulator and its impact on system performance.

The results show that the bandwidth is allocated more accurately using the controller, mainly when applications generate heterogeneous traffic in the system, obtaining an accuracy of $\pm 1\%$. On the other hand, it is

determined that latency does not significantly impact system performance.

Keywords— Bandwidth Regulation, QoS, SoC

Resumen

En la actualidad, al interior de un chip, es posible la integración de Elementos de Procesamiento (EPs) que colaboren en la ejecución de una tarea permitiendo cumplir con los requerimientos de Calidad en el Servicio (CeeS) establecidos (latencia y rendimiento). Para lograrlo, frecuentemente necesitan acceder a los recursos compartidos con diferente regularidad. En éste trabajo mostramos la arquitectura hardware de un regulador de ancho de banda para un Sistema de Interconexión tipo bus que permite su uso en proporción a los pesos de soporte de un árbitro del tipo Round-Robin. Lo anterior se logra modificando, en tiempo de ejecución, los pesos del árbitro. El comportamiento del regulador es analizado modelando el sistema en SystemC con precisión de ciclo de reloj. Las aportaciones de este artículo son:

- Diseño y modelado en hardware, de un regulador de ancho de banda para un sistema de interconexión tipo bus.
- Evaluación de la capacidad del regulador de asignar el ancho de banda de acuerdo a las proporciones establecidas.
- Determinación de la latencia derivada del uso del regulador y su impacto en el rendimiento del sistema.

* Autor de correspondencia

Los resultados muestran que el ancho de banda se asigna con mayor exactitud utilizando el regulador, principalmente cuando se tienen aplicaciones que generan tráfico heterogéneo en el sistema, obteniendo una precisión de $+/-1\%$. Por otro lado, se determina que la latencia no impacta significativamente en el rendimiento del sistema.

Palabras clave— Regulación de ancho de banda, QoS, SoC

I. Introducción

Los Sistemas en Chip (SeCs) son circuitos integrados que contienen un conjunto de EPs que colaboran entre sí para ejecutar diversas tareas. El Sistema de Interconexión (SI) es el encargado de la comunicación entre los EPs por medio de mensajes. Dada la capacidad de procesamiento que los EPs pueden alcanzar, generalmente la cantidad de tráfico generado por los EPs es muy grande, por lo que el rendimiento del sistema se ve mayormente afectado por el flujo de los mensajes dentro del SI. Diversos estudios han mostrado que el rendimiento de un SeC depende principalmente del SI que une a los EPs. Dally and Towles [1] indican que el rendimiento de los sistemas digitales se encuentra limitado por sus comunicaciones o interconexiones, no por su lógica o memoria, el mayor tiempo de reloj es utilizado en el retardo del cableado, no en el retardo de las compuertas. Poletti et al. [2] mencionan que conforme la complejidad de un SeC se incrementa, la arquitectura del sistemas de comunicación se convierte en el cuello de botella. Indican que el rendimiento de un sistema con múltiples procesadores depende más de un eficiente sistema de comunicación. Por su parte Benini and Bertozzi [3] plantean que mientras los elementos de procesamiento y almacenamiento se benefician del escalado de los dispositivos (puertas y celdas de memoria mas pequeñas), la energía de la comunicación global no se reduce, los retardos en la comunicación supera en gran medida el periodo de reloj del sistema. Ahmed et al. [4] establecen que el desarrollo de interconexiones en chip de alto rendimiento es de suma importancia para las tecnologías de computación paralela y de alto rendimiento. En [5] los autores indican que los sistemas de interconexión no pueden seguir el ritmo del ancho de banda y la complejidad de los recientes chips con múltiples procesadores heterogéneos. El impacto del tráfico en un SI toma mayor importancia si se considera que los SI tipo bus se pueden llegar a saturar fácilmente cuando el numero de PE's supera los ocho elementos [6]. En este caso el SI se encuentra cercano al 100% de su uso y el tiempo de espera en los registros de salida de las Interfaces de Red (IR) es alto.

Por otro lado, los EPs que colaboran para efectuar una tarea específica, tienen en diferentes momentos, diferen-

tes requerimientos del uso del SI. Es decir, que para poder cumplir con los requerimientos de CeeS que le fueron impuestos al sistema, los EPs tienen la necesidad de hacer un uso diferenciado del mismo. Lo anterior implica que al interior del sistema de interconexión debe existir un elemento que permita regular su uso evitando que EPs con una gran demanda del medio lo acaparen. Esta es una necesidad común a todos los tipos de sistemas de interconexión dentro de un SeC ya que de alguna forma u otra siempre se tiene la necesidad de compartir un recurso como se indica en la Tabla 1.

Tabla 1: Tipos de sistemas de interconexión en un SoC

Tipo	Recurso a compartir
Bus	Los PE's compiten para tener acceso al sistema de interconexión
Croosbar	Dos o más maestros compiten por acceder al uso de un esclavo
Redes-en-Chip	Dos o más flujos compiten por el acceso a un buffer

La regulación del ancho de banda en un sistema de interconexión como un objetivo para lograr cumplir con los requerimientos de CeeS ha sido investigado por diversos autores, un resumen de las técnicas utilizadas se muestra en [7].

En el presente trabajo se muestra una alternativa de solución para la regulación del ancho de banda en un sistema de interconexión tipo bus. Su implementación y desarrollo fue hecha en en SystemC con el cual se implementó un sistema de interconexión basado en bus utilizando la política de arbitraje Round-Robin con pesos (WRRRA) [1], aunque es posible utilizar cualquier otra política basada en pesos de soporte. Al sistema de interconexión se le incorpora el regulador que hemos diseñado. Este, en tiempo de ejecución, es capaz de evaluar el comportamiento del sistema de interconexión y modificar los pesos de soporte inicialmente asignados con el fin de encontrar los pesos que permitan asegurar una correcta distribución del uso del bus, de acuerdo a los porcentajes de uso inicialmente establecidos. Además, por medio de simulación, se muestra el comportamiento del uso del sistema de interconexión con la adición del regulador presentado. Los resultados muestran que utilizando el regulador es posible distribuir mejor el uso del bus en proporción al porcentaje de uso deseado por cada PE, logrando alcanzar una precisión de $+/-1\%$ con respecto a la proporción de los pesos inicialmente asignados.

El resto del trabajo se encuentra organizado de la si-

guiente manera: en la sección II se describe el sistema de interconexión con la política mencionada. En la sección III se presenta el regulador diseñado y su acoplamiento al sistema de interconexión. En la sección IV se muestran las pruebas realizadas al sistema utilizando el regulador y se compara con el sistema sin el uso del regulador. Finalmente en la sección V se mencionan las conclusiones del trabajo realizado.

II. Sistema de Interconexión

II.1. Sistema de Interconexión basado en bus

Un SI para un SeC basado en una topología tipo bus es mostrado en la Figura 1. Cada uno de los EPs que necesite conectarse al bus lo hace por medio de una IR asociada a el, ésta interfaz sirve como el elemento de unión entre el sistema de cómputo y el sistema de interconexión. La IR, por medio del módulo Transmisor de Red (TR) tiene un canal por donde transmite el mensaje que desea enviar al EP destino. El mensaje sólo será transmitido si el Controlador del Bus (CB) le otorga el acceso al sistema de interconexión. El proceso para otorgar el uso del bus a un EP depende de la política de arbitraje que se utilice, sin embargo, desde el punto de vista de las IR, cada vez que una de ellas requiera el uso del bus tendrá que pedirlo por medio de su línea de solicitud. La IR mantendrá la línea activa hasta el momento en que el medio le sea asignado por el CB, el cual le indica por medio de la señal de concesión que puede hacer uso del bus. A partir de este momento, la IR tendrá el control del bus el tiempo necesario para transmitir su mensaje, esto lo logra manteniendo la línea de sostener activa la cantidad de ciclos de reloj que se requieran para transmitir el mensaje. La IR libera el bus cuando desactiva la línea de sostener, dando la oportunidad a que otro EP tome el control del bus.

En cualquier sistema de interconexión se evidencia la necesidad del uso de políticas de arbitraje que tienen como objetivo resolver conflictos cuando dos o mas dispositivos intentan acceder a un mismo recurso. Entre otros problemas que se intentan resolver Jain et al. [8] mencionan los siguientes:

- *Congestion* - Se da cuando muchos puertos de entrada se encuentran solicitando un mismo puerto de salida.
- *Starvation* - Es un tipo de injusticia en la cual no todos los puertos de entrada tienen las mismas oportunidades de acceder al puerto de salida.
- *Deadlock* - Un puerto de salida no puede ser alcanzado por un puerto de entrada debido a que este se encuentra esperando por otro puerto de entrada para liberar el recurso.

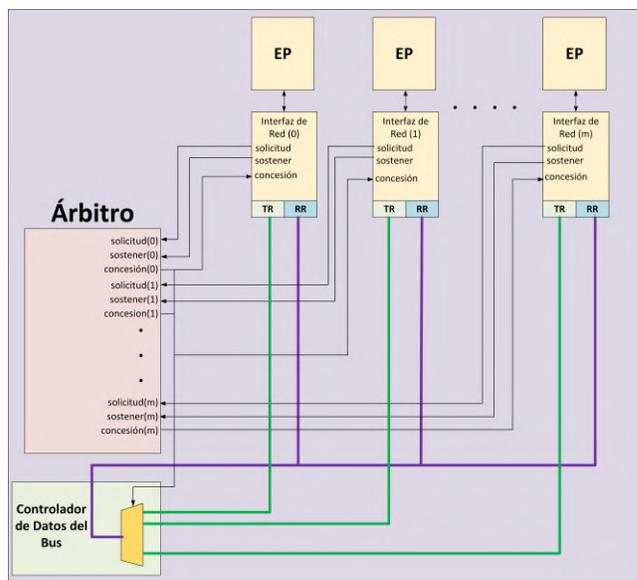


Figura 1: Configuración básica de un Sistema de Interconexión tipo bus

- *Livelock* - Los paquetes de un puerto de entrada están en movimiento pero no pueden alcanzar el puerto deseado de salida.
- *Header Blocking* - Este problema se presenta principalmente en los routers. Consiste en que un puerto de entrada de un router se encuentra con el buffer lleno debido a que el sistema de arbitraje le esta dando preferencia a otro puerto. Si en el router anterior a este hay un paquete que viene a este puerto, el paquete no puede avanzar porque el buffer de entrada esta lleno y retiene a los paquetes que vienen detrás de el aunque vayan dirigido a otro puerto.

Para resolver los problemas anteriormente citados, se han realizado una serie de propuestas específicas para cada uno de ellos, una recopilación de estas puede ser encontrado en Ibarra-Delgado et al. [9]

II.2. Árbitro Basado en pesos de soporte

El regulador que se propone en este documento puede ser utilizado en árbitros que contemplen diferentes políticas de arbitraje. Con la finalidad de mostrar de forma clara la eficiencia del mismo, se establece como árbitro de prueba el árbitro propuesto por Dally and Towles [1].

El árbitro *WRR* tiene como propósito principal controlar el grado en que un solicitante puede ser atendido con respecto a otros. A cada uno de los solicitantes le es asignado un peso w_i que indica la fracción de uso del recurso que le será otorgado. El peso indica el número de ciclos de reloj que le serán otorgados al solicitante, este valor es establecido por una entidad superior en relación

con la necesidad de uso del SI. El peso acumulado de todos los recursos esta dado por $W = \sum_{i=0}^{N-1} w_i$ donde N es la cantidad de solicitantes, de tal modo que la fracción de tiempo asignada al solicitante S_i queda dada por $f_i = \frac{w_i}{W}$. Mientras mayor sea el valor de w_i mayor sera la fracción de tiempo que se le asignará al solicitante S_i . En la Figura 2 se puede observar la arquitectura del árbitro WRR. En ésta política de arbitraje, se le asigna un peso inicial a cada solicitante, solamente los solicitantes que tengan un peso diferente de cero serán los que puedan entrar a la ronda de arbitraje. Una vez que un solicitante obtuvo el acceso al recurso, durante cada ciclo de reloj que mantenga el uso del mismo le será descontado una unidad de su peso, cuando éste peso llegue a cero, ya no podrá contender por el recurso hasta que el peso le sea recargado. El peso le es recargado a todos los solicitantes simultáneamente cuando el peso de todos ha llegado a cero.

La principal ventaja de este tipo de árbitro consiste en la posibilidad de controlar el porcentaje de uso de un recurso de acuerdo al peso asignado. La principal desventaja es que cuando un solicitante agotó su peso ya no puede tener acceso al recurso aún y cuando el recurso se encuentre desocupado, lo anterior puede suceder porque otros solicitantes aún cuentan con cierta cantidad de peso y estos solamente son recargados hasta cuando el peso de todos los solicitantes es cero. Esta forma de operar del árbitro puede provocar que el SI tenga ciclos de ocio lo cual impacta directamente en su rendimiento.

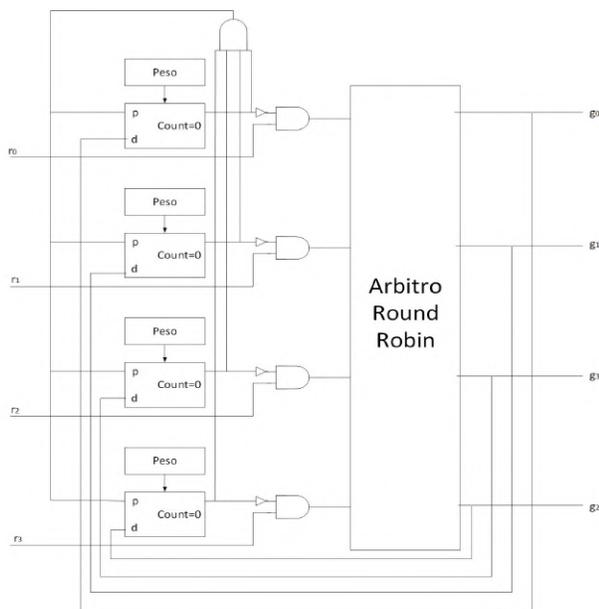


Figura 2: Arquitectura de un árbitro WRR

III. Regulador de acceso al Bus

WRR es un árbitro que cumple con los requerimientos de regulación establecidos por los pesos de soporte, pero solo bajo ciertas condiciones: primera, que el tamaño de los paquetes que son transmitidos sea homogéneo de tal modo que puedan ser divididos exactamente en el tiempo (peso) que les fue asignado; segunda, que la diferencia en los tamaños de paquetes generados por los diferentes EPs no sea significativo. En aplicaciones reales las condiciones anteriores son difíciles de alcanzar, en la suite de vectores de prueba que presentan Liu et al. [10] se muestra con claridad que en aplicaciones reales como la transformada rápida de fourier, la codificación/decodificación de video entre otras, las tramas que se generan, no solo son de tamaño diferente entre la diferentes tareas que componen la aplicación, sino también en una tarea específica dependiendo de su contexto.

Dado lo anterior se propone insertar un regulador en el SI, que permita la verificación del uso del bus para cada uno de los EPs dada de una ventana de tiempo establecida. El regulador tiene la capacidad de reajustar en tiempo de ejecución los pesos originalmente establecidos con la finalidad de que el sistema cumpla con las proporciones de uso indicadas inicialmente. En la Figura 3 se puede observar el sistema propuesto.

Para que el regulador pueda operar es necesario cargar un valor en la *Ventana de tiempo* cuyo valor máximo se encuentra en el orden de un millón de ciclos de reloj, también deben de ser inicializados los *Pesos Iniciales* de cada EP en términos de porcentaje de utilización del bus deseado para cada uno. El valor introducido en la *Ventana de tiempo* indica el número de ciclos de reloj que tendrán que transcurrir para evaluar el estado del uso del bus y realizar los ajustes necesarios.

La Máquina de Estados Finitos (FSM) que controla el regulador inicia su operación al recibir la señal de *start* como se puede observar en la Figura 4. La primera acción de la FSM es tomar cada uno de los porcentajes iniciales que fueron cargados en los registros denominados *Pesos Iniciales*. Estos valores se convierten en una dirección de memoria la cual es utilizada por la FSM para apuntar a la denominada *Memoria de Pesos* que es una memoria ROM donde están almacenados los pesos correspondientes al porcentaje indicado, este valor está en concordancia con el valor asignado en la *Ventana de tiempo*. Por ejemplo, si en la *Ventana de tiempo* se tiene una cantidad de 200,000 ciclos, entonces en la dirección uno de la *Memoria de Pesos* se encontrará almacenado el valor 2,000 que corresponde al 1% del tamaño de la ventana, en la dirección 2 se encontrará almacenado al valor 4,000 correspondiente al 2% del tamaño de la ventana y así sucesivamente. La FSM activa las señales necesarias para cargar los pesos en el WRR y que comience su operación.

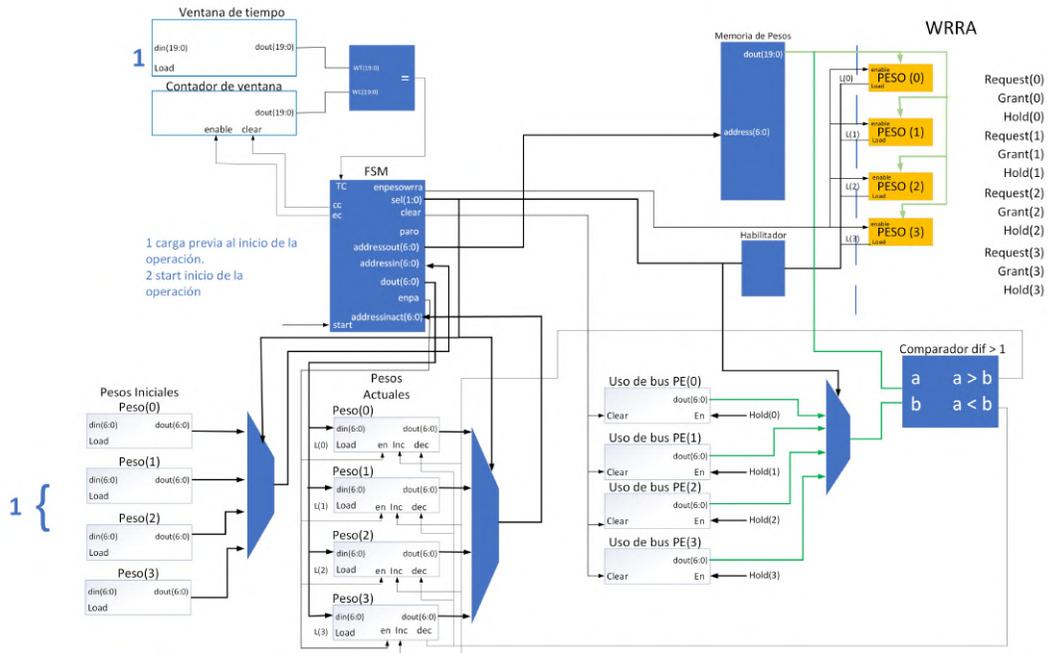


Figura 3: Arquitectura del regulador

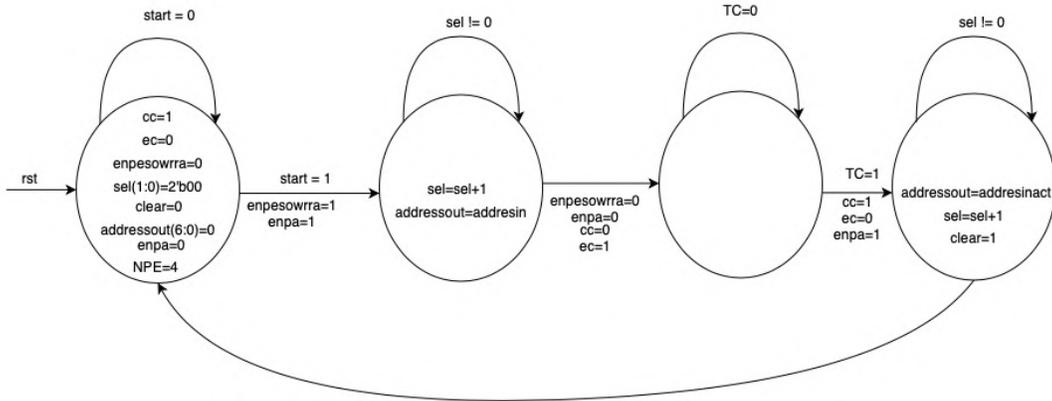


Figura 4: Máquina de Estados Finitos que controla el regulador

La FSM borra y habilita el denominado *Contador de ventana* que es un contador ascendente cuyo valor es verificado cada ciclo de reloj, cuando este contador alcanza el valor establecido en la *Ventana de tiempo* le indica a la FSM que es momento de evaluar el uso real del bus por cada uno de los PEs.

Por otro lado al inicio de la operación de la FSM los denominados registros *Uso de bus* son inicializados en cero, éstos registros son incrementados en uno cada ciclo de reloj mientras la señal de *sostener* del WRRa este activa para un PE, con esto se tiene una cuenta exacta del número de ciclos de reloj que un PE uso el bus durante una ventana de tiempo, cuando esto sucede la FSM compara para cada PE el uso del bus real, contra el uso del bus definido en la *Memoria de Pesos* por los pesos ini-

ciales, si la diferencia es mayor al 1 % los denominados registros *Pesos Actuales* son modificados incrementando o decrementando por uno su peso anterior y estos pesos son ahora cargados en los registros de pesos del WRRa para la operación en la siguiente ventana de tiempo.

Cuando una ventana de tiempo se agota la FSM calcula los nuevos pesos de acuerdo el procedimiento señalado. Cuando esto ocurre la FSM espera a que termine la transacción actual y procede a realizar el cálculo. El tiempo invertido en el cálculo de los nuevos pesos (latencia) depende del número de EP que hay en el SI. Sin embargo, por cada EP la FSM toma el valor del peso actual del EP el cual se convierte en el índice a la *Memoria de Pesos* donde se encuentra el valor correspondiente al numero de ciclos esperados a utilizar para un porcentaje de uso

dado, este valor es comparado con el porcentaje de uso real y dependiendo de la comparación el peso actual (el índice a la memoria de pesos) es incrementado o decrementado en 1, es decir se modifica el índice del peso. Para realizar este procedimiento por cada EP solo son necesarios 4 ciclos de reloj, en el caso de que se tengan 8 EPs el tiempo de cálculo es de solo 32 ciclos de reloj.

IV. Pruebas y Resultados

En estudios anteriores [9] se ha comprobado que el principal problema con la asignación proporcional a los pesos de soporte asignados a cada EP se encuentra cuando existen las siguientes condiciones:

- El sistema se encuentra saturado o cerca de su estado de saturación.
- Los patrones de tráfico generados por los EPs son heterogéneos.

Tomando en cuenta lo anterior, para las pruebas realizadas se utilizan los patrones de tráfico generados por la suite que presentan Liu et al. [10] para aplicaciones reales donde se han identificado los siguientes tipos de patrones de tráfico:

- **Tráfico con paquetes de tamaño pequeño:** Son paquetes cuyas tramas encuentran en el orden de las unidades. Como ejemplo se encuentra el algoritmo FFT-1024 complex que genera una carga útil de entre cinco y siete *phits* (unidad de transmisión física del canal de comunicación en un SI).
- **Tráfico con paquetes de tamaño mediano:** este tipo de paquetes tienen una carga útil en el orden de las decenas de *phits*. En la suite utilizada el algoritmo FPPPP genera este tipo de tráfico.
- **Tráfico con paquetes de tamaño grande:** Son paquetes con carga útil en el orden de cientos de *phits*. El algoritmo de codificación de video H264 de la suite genera este tipo de paquetes.

El tamaño de la ventana debe de ser establecido de tal modo que no afecte significativamente el rendimiento del sistema. Mientras menor sea este tamaño la regulación tomará menos tiempo a cambio de una latencia mayor. En la pruebas aquí realizadas se estableció un tamaño de ventana de 200,000 ciclos, se puede observar que con este valor en tan solo seis ventanas de tiempo el regulador se encuentra muy cerca del estado deseado y que su latencia es mínima.

En la primer prueba realizada se ejecutan simultáneamente 3 algoritmos del tipo FFT-1024 complex con la misma relación de pesos. En la Figura 5 se puede observar que el sistema se encuentra saturado, sobre el 100% de utilización y que el comportamiento del bus con la

política de arbitraje WRRR es estable muy cercana a los valores de porcentaje de uso establecidos. Con el uso del regulador los pesos son ligeramente modificados entre ventana y ventana de muestreo sin alejarse demasiado de los valores inicialmente establecidos. En esta prueba los valores de uso del bus por cada EP son muy similares, cercanos al 33% en ambos casos, por lo que el uso del regulador no presenta ninguna ventaja.

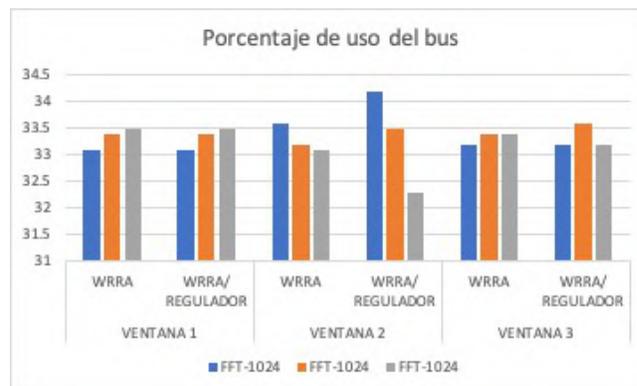


Figura 5: tres aplicaciones iguales generando el mismo tipo de tráfico

En la segunda prueba realizada se ejecutan los mismos tres algoritmos de la prueba anterior pero ahora con una relación de pesos 1/1/2 lo que significa que los paquetes generados por los dos primeros algoritmos tendrán un porcentaje del 30% mientras que el tercero del 40% los resultados mostrados en la Figura 6 muestran que la política de arbitraje WRRR no permite alcanzar la relación deseada, esto sucede principalmente por una razón: cuando un PE consumió el tiempo que tiene asignado para uso del bus, ya no tiene posibilidad de usarlo aun y cuando los otros PE's no lo estén necesitando, lo cual le impide llegar a su cuota asignada. Se puede observar que con el regulador al ser modificado el peso, eventualmente el uso del bus se acerca mas a los valores establecidos originalmente.

Finalmente se efectúa una prueba donde se ejecutan tres algoritmos que generan tráfico heterogéneo: FFT-1024 complex, FPPPP para síntesis química y Codec de video H264. La relación de uso del bus es la misma para los tres algoritmos. En la Figura 7 se puede observar que el sistema está saturado y que la política de arbitraje WRRR le otorga una mayor uso del bus al algoritmo que genera paquetes de mayor tamaño, lo cual va en detrimento del algoritmo con paquetes de menor tamaño que lo deja muy alejado del valor esperado de uso del bus. Al insertar al regulador y poder modificar los valores de los pesos se puede observar que eventualmente los tres algoritmos se acercan al valor requerido de uso del bus para cada uno.

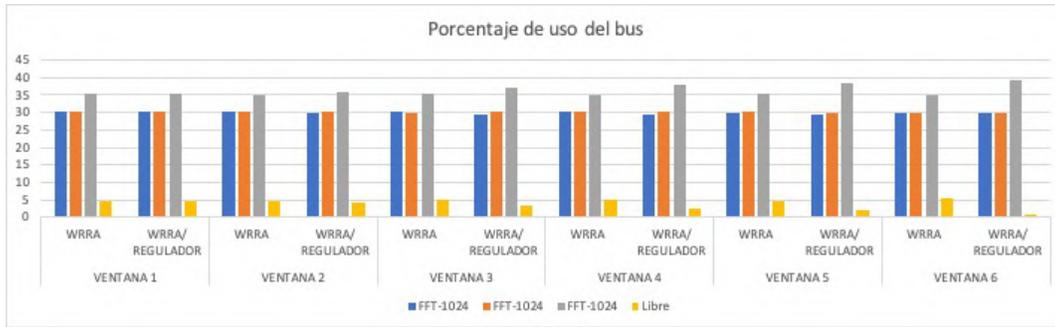


Figura 6: tres aplicaciones iguales generando diferente cantidad de tráfico

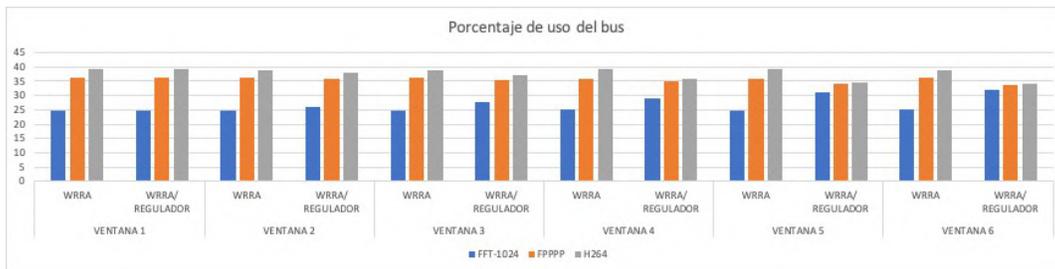


Figura 7: tres aplicaciones diferentes generando diferente cantidad de tráfico

V. Conclusiones

De las pruebas que fueron realizadas al sistema se puede observar que el regulador contribuye a una mejor asignación del ancho de banda dependiendo del tipo de tráfico que existe en el SI. Cuando el tráfico es homogéneo y la asignación de ancho de banda es igual para todas las aplicaciones, el sistema se comporta de forma similar con o sin la adición del regulador, siendo aún más estable sin la adición del mismo. La principal ventaja de utilizar el regulador se evidencia cuando el tipo de aplicaciones que se ejecutan en los EPs dentro del sistema embebido tienen relaciones de uso diferentes y el tráfico de las aplicaciones es heterogéneo. En este caso, las pruebas muestran que mientras el sistema con la política de arbitraje sin regulador se encuentra lejos de alcanzar la relación deseada, hasta en un 8 %, cuando el regulador es agregado esta relación alcanza un $\pm 1\%$ del valor deseado. Lo anterior permite acercar a las aplicaciones que corren en el SoC a cumplir con sus requerimientos de CeeS.

Por otro lado, se pudo determinar que la latencia para calcular los nuevos pesos en cada ventana de muestreo es de apenas de unas decenas de ciclos. En el caso de un SoC con un *Sistema de interconexión* basado en bus donde el número de EPs es reducido, no mayor a 10 en la mayoría de los casos, esta latencia no representa más del 0.02 % del tamaño de la ventana establecido. Lo anterior nos permite concluir que el rendimiento del sistema no se ve afectado significativamente por la integración del

regulador. En el diseño se estableció un tamaño máximo de ventana de 1 millón de ciclos en el futuro es importante estudiar el efecto que tiene este tamaño en términos del rendimiento del sistema. Es importante mencionar que el umbral de diferencia puede tener mayor precisión, para lograrlo solo es necesario aumentar el número de valores en la Memoria de Pesos, en términos de rendimiento el sistema no se ve comprometido. Además, en el futuro se pretende evaluar otras estrategias para realizar esta regulación, poniendo especial énfasis en la co-diseño hardware-software.

Referencias

- [1] William Dally y Brian Towles. *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003. ISBN: 0122007514.
- [2] Francesco Poletti y col. «Performance Analysis of Arbitration Policies for SoC Communication Architectures». En: *Des. Autom. Embedded Syst.* 8.2 (jun. de 2003), págs. 189-210. ISSN: 0929-5585. DOI: 10.1023/B:DAEM.0000003962.54165.5c. URL: <https://doi.org/10.1023/B:DAEM.0000003962.54165.5c>.
- [3] L. Benini y D. Bertozzi. «Network-on-chip architectures and design methods». English. En: *IEEE Proceedings - Computers and Digital Techniques* 152 (2 mar. de 2005), págs. 261-272. ISSN: 1350-

2387. URL: <https://digital-library.theiet.org/content/journals/10.1049/ip-cdt-20045100>.
- [4] Khaled E. Ahmed, Mohamed R. Rizk y Mohamed M. Farag. «Overloaded CDMA crossbar for network-on-chip». En: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.6 (2017), págs. 1842-1855. ISSN: 10638210. DOI: 10.1109/TVLSI.2017.2664660.
- [5] Ahmed Ben Achballah, Slim Ben Othman y Slim Ben Saoud. «Problems and challenges of emerging technology networksonchip: A review». En: *Microprocessors and Microsystems* 53 (2017), págs. 1-20. ISSN: 0141-9331. DOI: <https://doi.org/10.1016/j.micpro.2017.07.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0141933116303593>.
- [6] L. Benini y G. De Micheli. «Networks on chips: a new SoC paradigm». En: *Computer* 35.1 (ene. de 2002), págs. 70-78. ISSN: 1558-0814. DOI: 10.1109/2.976921.
- [7] Salvador Ibarra-Delgado y col. «A Bandwidth Control Arbitration for SoC Interconnections Performing Applications with Task Dependencies». En: *Micromachines* 11.12 (2020). ISSN: 2072-666X. DOI: 10.3390/mi11121063. URL: <https://www.mdpi.com/2072-666X/11/12/1063>.
- [8] Kunj Jain y col. «Problems encountered in various arbitration techniques used in NOC router: A survey». En: *2015 International Conference on Electronic Design, Computer Networks and Automated Verification, EDCAV 2015*. 2015, págs. 62-67. ISBN: 9781479962082. DOI: 10.1109/EDCAV.2015.7060540.
- [9] Salvador Ibarra-Delgado. «Diseño de sistemas embebidos reconfigurables empleando elementos de interconexión en Chip definidos por Software». En: *Helvia:: Repositorio Institucional de la Universidad de Córdoba* (2021). URL: <https://helvia.uco.es/bitstream/handle/10396/21536/2021000002283.pdf?sequence=1&isAllowed=y>.
- [10] W. Liu y col. «A NoC Traffic Suite Based on Real Applications». En: *Proceedings of the 2011 IEEE Computer Society Annual Symposium on VLSI*. IEEE, 2011, págs. 66-71.