Arquitectura genérica de una red en chip de enrutamiento unidireccional en FPGA

Jorge Ernesto López Arce Delgado
CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
je.lopezarce.d@gmail.com

Juan José Raygoza P.
CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
juan.raygoza@cucei.udg.mx,

Susana Ortega Cisneros
Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
susana.ortega@gdl.cinvestav.mx

Jorge Rivera D.
Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
riveraj@gdl.cinvestav.mx

Pablo Moreno Villalobos
Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
pmoreno@gdl.cinvestav.mx

Resumen

El constante aumento de los componentes que contiene un sistema *on-chip* ha incrementado la complejidad de comunicación entre los elementos de procesamiento (EPs) del sistema. Un recurso utilizado para disminuir la complejidad es el diseño de enrutamiento de conexiones (cableado), el cual ha sido suficiente para interconectar algunos EPs, dicho diseño se conoce como redes en chip o por sus siglas en inglés NoC (*Network on Chip*), de manera alternativa, enrutar paquetes permite una mayor escalabilidad de las redes, tener una latencia aceptable y una utilización moderada de

área. Sin embargo, las redes en chip (NoC) suelen ser implementadas en tecnologías rígidas y deterministas como los ASIC (Circuito Integrado de Aplicación Específica), limitando la flexibilidad, arquitectura y modularidad que ofrece una NoC de enrutamiento de paquetes. Este trabajo propone una arquitectura de una red en chip de switcheo o enrutamiento unidireccional utilizando un router genérico para topología de mariposa, de enrutamiento de paquetes, implementado en una FPGA de la familia Xilinx. Donde el diseño permite enviar paquetes desde 16 puntos de origen, hacia 16 puntos de destino, así como la flexibilidad de enviar paquetes de diferentes tamaños, divididos en flits. Este diseño tiene como resultado una arquitectura compacta, permitiendo dejar el mayor espacio posible para los EPs.

Palabra(s) Clave(s): arquitectura de router, control de flujo, FPGA, NoC, redes en chip.

1. Introducción

La característica principal de la tecnología de las FPGA (Field programmable Gate Array) es la posibilidad de ser reconfiguradas electrónicamente, contrario a otros circuitos integrados, como los ASICs, donde el hardware tiene funciones rígidas y deterministas. Esta flexibilidad permite a las FPGAs ser configuradas para aplicaciones computacionales como son el procesamiento de señales, el procesamiento de imágenes y la criptografía, típicamente tienen diseños con mejor desempeño en comparación con CPUs (*Central Processing Unit*) tradicionales[1,2]. La tecnología ha tenido una evolución tal, que en un futuro las FPGAs podrían llegar a tener millones de LUTs (*Look Up Tables*) dando la posibilidad de diseñar arquitecturas con muchos procesadores trabajando en paralelo, generando más lógica de circuitos, y a medida que la complejidad incrementa, la lógica de circuitos entre los procesadores hace que los diseños RTL (*Register File Level*) tradicionales sean ineficientes[1]. Para lograr aprovechar la evolución de las FPGAs es necesario dar énfasis al diseño y manejo de los recursos en un nivel de abstracción mayor, ejemplo de ello, sería considerar el diseño a nivel de función, es decir, si se desea implementar procesadores en paralelo,

pensar en su programabilidad para operar con otros simultáneamente, en vez de solo poder programar conexiones entre ellos. Un nivel de abstracción mayor presenta limitaciones en el uso de buses tradicionales y esquemas de interconexión punto a punto, en términos de la escalabilidad y complejidad del chip [3]. Como consecuencia, hoy en día se observa una tendencia de crear sistemas con arquitecturas complejas, velocidades de reloj mejoradas y aumento de la densidad de lógica en los chips[1]. Un nuevo paradigma de diseño que emerge son las Redes en Chip o NoC, por sus siglas en inglés, las NoC son una solución eficiente y económica para los requerimientos de comunicación aplicados en un chip. Las NoCs pueden ser diseñadas y construidas para ocupar menos área que un diseño conectado espacialmente y ofrece mejor eficiencia que un bus[4]. La red en si es una colección de interruptores conectados entre ellos usando canales cableados. Permitiendo a los EPs enviar tráfico de información a la red y llegar a su destino a través de *switches* [5].

El diseño de una red está definida por la forma en que se implementan tres elementos fundamentales que son la topología, enrutamiento y control de flujo de datos. [6]

La interconexión de red se implementa con una colección de nodos enrutadores distribuidos mediante canales compartidos. El patrón de conexión de estos nodos definen la topología de la red, y la topología de una interconexión de red es parecido a una carretera de autos, donde en los canales (como carreteras) transportan paquetes (los autos) de un nodo enrutador a otro (intersecciones), es decir es el camino por el cual los paquetes viajan. Es necesario definir el tipo de topología antes de elegir qué tipo de enrutamiento será seleccionado. El mensaje enviado por la red es entregado entre las terminales haciendo “saltos” a través de los canales y nodos compartidos desde su terminal-origen hasta su terminal-destino. Una vez que la topología es definida, aún falta seleccionar el camino que el mensaje va a tomar (la secuencia de salto en los nodos) en la red para llegar a su destino. El enrutamiento determina cuál es el camino que toma el mensaje, donde la extensión del camino impacta directamente en la latencia del mensaje a través de la red. El control de flujo de datos determina como

serán asignados los diferentes recursos de la red, ancho de banda del canal y estado de control para que los paquetes transiten. Un buen método de control de flujo debe repartir estos recursos de una manera eficiente facultando a la red de tener un alto porcentaje de su ancho de banda ideal y entrega de paquetes con una baja latencia predecible [5,6,7].

2. Desarrollo

Este trabajo presenta la arquitectura de una NoC con topología tipo mariposa implementada en una FPGA, la arquitectura se ha diseñado en lenguaje Verilog bajo el estándar IEEE 1364-2001. La tarjeta de desarrollo que se utilizó para evaluar este diseño es una kit Zynq-700 SoC ZC702, la cual cuenta con una FPGA XC7Z020, arquitectura basada en los SoCs de la familia de Xilinx, la cual tiene como características principales 106,400 Flip-Flops, 53,200 LUTs (*Look Up Tables*), 200 pines de I/O, 32 BUFG (Buffers globales), 140 bloques de RAM36/FIFO, 280 bloques de RAMB16, 220 DSPs. Esta tarjeta puede integrar diferentes elementos al diseño, muestra de ello es el elemento principal, pues esta tarjeta contiene dos CPUs de arquitectura ARM Cortex–A9 ubicado en la sección del sistema de procesamiento, la cual se puede habilitar para integrarse con módulos diseñados en la sección de lógica programable[8].

La arquitectura que se propone cuenta con 16 terminales-origen y 16 terminales-destino, en esta topología, desde un punto de vista de la terminal-origen hacia los posibles destinos se aprecia forma de árbol, donde en cada nivel de dicho árbol se encuentran nodos de *switcheo* (ver Fig. 1), estos nodos no envían ni reciben paquetes, solo los transmiten transmiten, hasta que lleguen a las terminales-destino. Los canales van en sentido de izquierda a derecha, como lo indican las flechas, donde los elementos de entrada se encuentran a la izquierda. En la Fig. 1 se observa que el diseño tiene 2 niveles o etapas de nodos de *switcheo*, y cada nodo tiene 4 canales de

entrada y 4 canales de salida de un ancho de 18 bits cada canal, siendo dos bits de cabecera.

El algoritmo de enrutamiento (routing) que se emplea en esta red en chip de switcheo es de tipo “etiqueta destino”. Donde los bits de destino son usados para seleccionar la salida de cada nodo. Se tiene dos niveles/ etapas de nodos, por lo tanto la dirección está constituida por cuatro bits, siendo el primer par los bits más significativos los que definen la salida del primer nivel de nodos (de izquierda a derecha), una vez que se “usan/leen” los dos bits de dirección en la primera etapa de nodos, se hace un corrimiento a la izquierda de dos posiciones en los bits de la dirección, quedando el segundo par de bits colocados para ser “usados/leídos”.

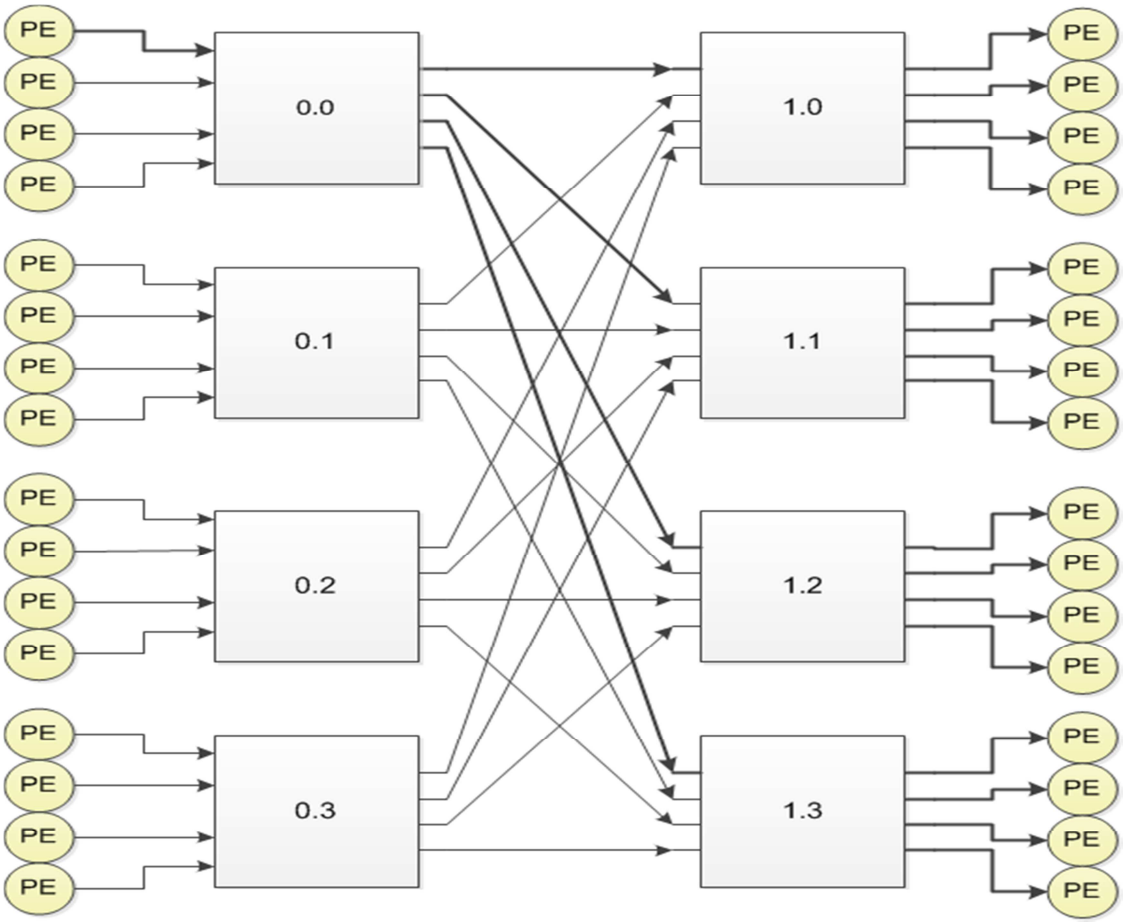

Fig. 11. Topología tipo mariposa.

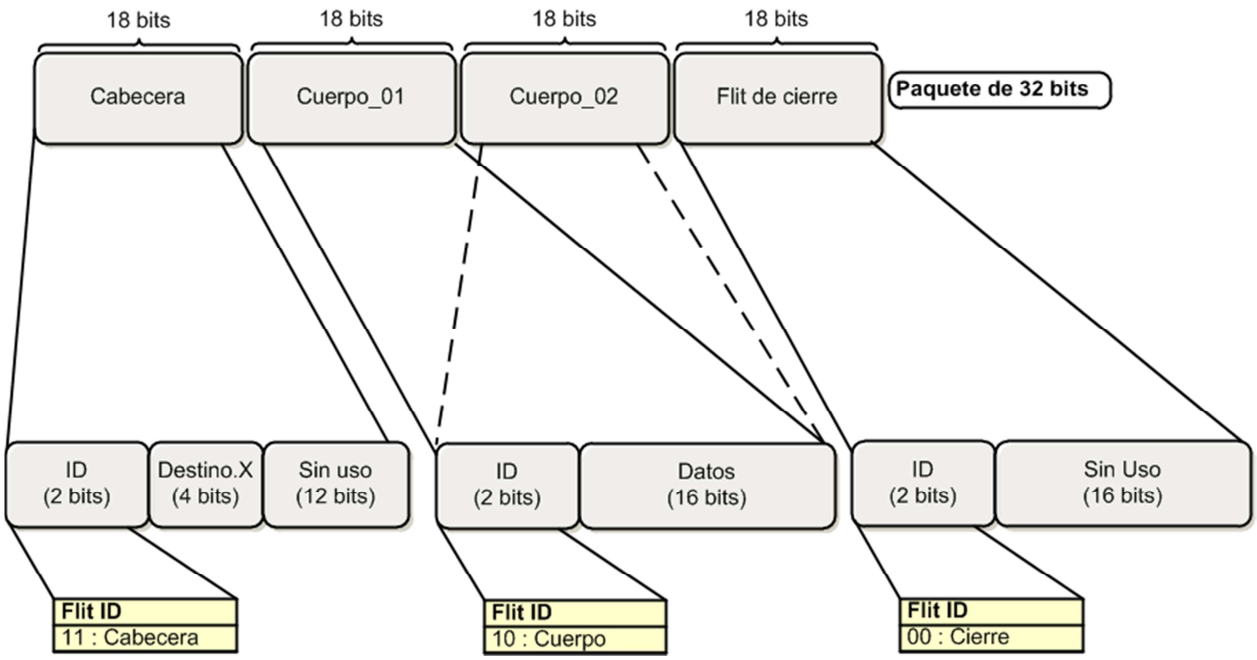

Fig. 12. Formato del paquete y formato de flit.

El control de flujo de datos en esta red de switcheo permite enviar entre 32 y 240 bits de datos. Se tiene un envío de 16 bits físicos por ciclo. Para poder enviar paquetes de ese tamaño se utiliza un protocolo, que permite dividir los paquetes en elementos más pequeños llamados flits, estos flits son de 18 bits, al igual que los canales de la red. Al ser un protocolo, se establecen diferentes tipos de flits para poder enviar no solo los datos del paquete sino también la información del ruteo, como bits de identificación del flit y la dirección de envió. Existen tres tipos de flits: de cabecera, de cuerpo y flits nulos (*Idle*). Los flits de cabecera tienen los primeros dos bits (bits más significativos) como bits de identificación,. Se puede observar en la Fig. 2, que el flit de cabecera contiene la información del destino, los flits de cuerpo los datos que conforman el paquete, y el flit nulo indicando el fin del envío del paquete así como el cese del uso del canal.

Diseño del router

Cada nodo de esta red es un router capaz de recibir paquetes en sus entradas, determinar su destino basado en su algoritmo de enrutamiento, y pasar los paquetes a

la salida correspondiente. La ruta de los datos dentro del router consiste en cuatro entradas de 18 bits, cuatro multiplexores 4:1 de 18 bits, cuatro módulos de desplazamiento (que realiza el desplazamiento de la dirección destino en cada nivel) y cuatro registros de salida de 18 bits (ver Fig. 3).

Los flits llegan en cada ciclo de reloj al registro de entrada y son enrutados a los cuatro multiplexores. Cada multiplexor está asociado a un módulo llamado “asignador”, donde se examinan los bits que definen el tipo de flit recibido y los siguientes dos bits, en el caso de que sea flit de cabecera, establece el *switcheo* correspondiente, así como envía la señal de habilitación al módulo de corrimiento para que modifique el campo de la dirección de destino y quede listo para el siguiente nivel disponible. Los flits de cuerpo pasan sin cambios.

El control del router está esencialmente en los cuatro “asignadores” asociados a cada salida, y a su vez, cada uno controla el selector del multiplexor y el módulo de corrimiento correspondiente. Cada “asignador”, destina un puerto de salida con un puerto de entrada. Un “asignador” consta internamente de cuatro partes iguales, cada uno dividido en tres secciones, en la primera se realiza la decodificación, en la segunda el arbitraje y en la tercera se define si se mantiene el canal o no. En la decodificación se toman los cuatro bits más significativos de entrada (dos del tipo de flit y dos de dirección) , cada decodificador genera dos señales, una se activa cuando el flit es de cabecera y el otro cuando los siguientes dos bits coinciden en representar el número de puerto de salida.

Estas señales indican que el flit de entrada hace la petición de usar el canal, así como cierto puerto de salida para pasar por ahí un paquete de datos. Cuando llega un flit de cuerpo el decodificador activa y envía una señal de retención a la tercera sección, indicando que el canal se mantiene pues el paquete aún no termina de pasar. La segunda sección es donde actúa el módulo de arbitraje de cuatro entradas y de prioridad fija, prioridad que otorga al canal de menor valor, es decir que ‘entrada_0’ es la de mayor prioridad y la ‘entrada_3’ la de menor prioridad.

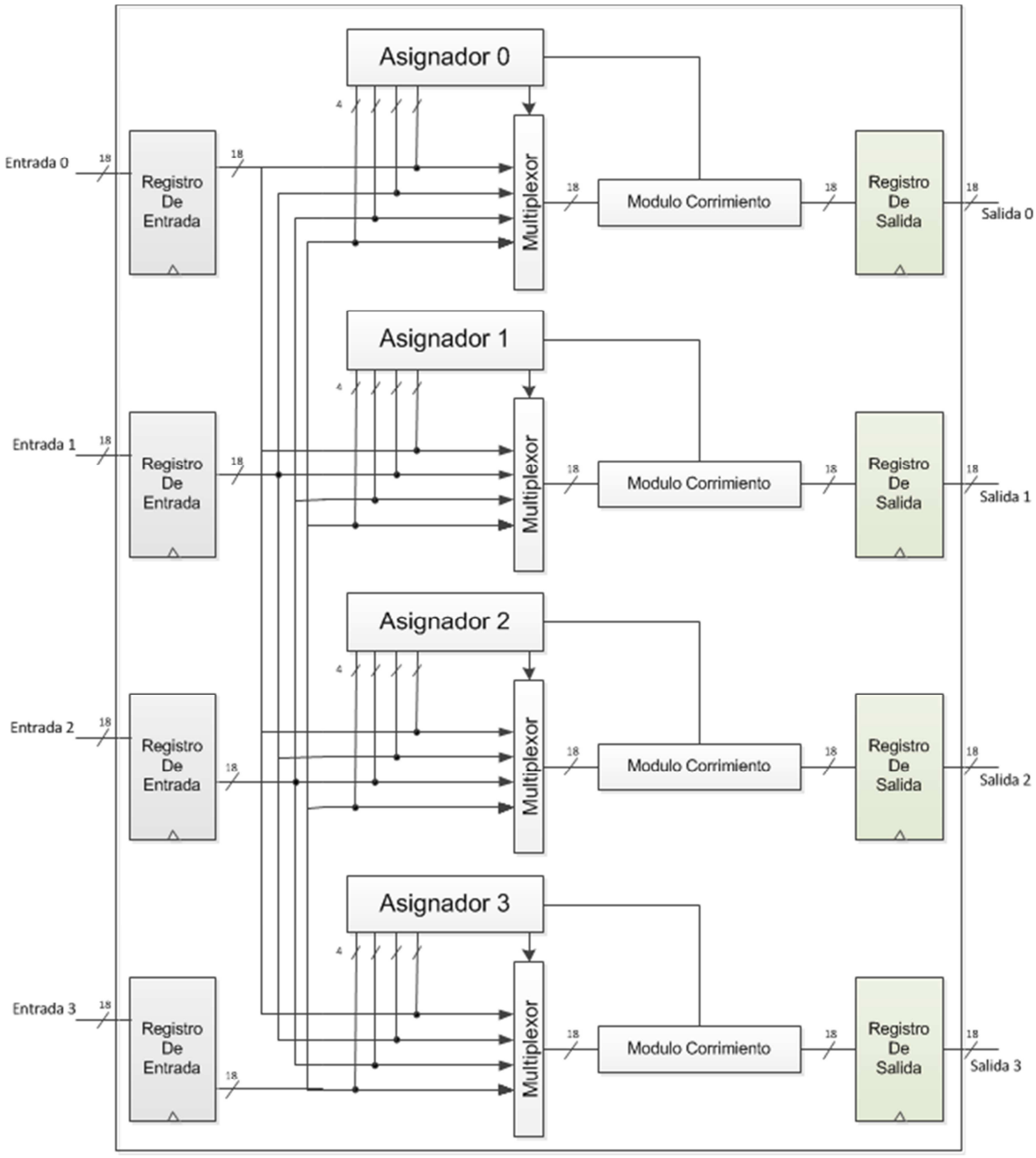

Fig. 13. Módulos internos del router.

El árbitro permite cuatro peticiones y genera cuatro señales de concesión de aprobación. Si el puerto de salida que se pide está disponible, el árbitro activa la señal de concesión al puerto de entrada que haya hecho primero la petición y tenga prioridad

mayor. Una vez que se asegura la señal de asignación, provoca que se garantice la señal de selección que indica la salida que tendrá el multiplexor. Si una señal de asignación se ratifica, indica que un flit de cabecera está pasando por el nodo, por lo tanto tiene que ser modificado por el módulo de corrimiento y envía la señal de habilitación hacia ese módulo. La última sección del "asignador" es el de retención del canal, el cual fija el camino entre el puerto de salida y entrada hasta que pase el paquete completo. También se tiene una señal de disponibilidad, la cual es deshabilitada cuando se tiene un canal activo en el ciclo anterior, indicando que aún no termina de pasar el paquete, y evita que se asigne el puerto a una nueva cabecera.

3. Resultados

En el análisis de desempeño, se consideran tres parámetros, latencia, costo de ocupación y rendimiento. La latencia es el tiempo que tarda un paquete en pasar a través de la red, el rendimiento es la cantidad de bits por segundo que la red puede transportar desde la entrada a la salida y el costo de ocupación se mide en el total de recursos de hardware utilizados por el diseño final en la FPGA.

Para evaluar el diseño se realizó un análisis de envío de paquetes de 32, 64, y 240 bits, se utilizan memorias precargadas con los datos que conforman los paquetes, se empleo un módulo externo a la arquitectura de la NoC el cual se encuentran las memorias (EPs) y se encarga de leer en cada ciclo de reloj un registro de memoria y es enviado a una entrada de la red, las memorias tienen paquetes armados para ser enviados a la red, dichos paquetes tienen la información de direccionamiento para que tomen rutas diferentes, el orden de envío es ascendente en los nodos de entrada, iniciando en la entrada 0 y descendente en el orden de salida, iniciando en la entrada 15, esto es que los paquetes enviados por la entrada 0 se envían a la salida 15, los que son enviados por la entrada 1 llegan al nodo de salida 14, así sucesivamente.

En la Fig. 4 se muestra la latencia del envío de 16 paquetes a distintas salidas (oR) con un ciclo de reloj de 2ns, usando todos los nodos de entrada, salida y canales de la red.

La Fig. 5 muestra el intervalo de 6 ns que hay entre la recepción de un paquete y otro en diferentes nodos de salida, revelando el tiempo de latencia en condiciones ideales donde siempre hay tráfico de paquetes en la red.

La intención es mostrar la fiabilidad de envío y recepción de paquetes en el mejor y peor caso, es decir, enviar y recibir los paquetes con la menor cantidad de bits (32 bits) en el periodo de tiempo más largo (50ns), así como envío y recepción de los paquetes de datos más grandes (240 bits) en el periodo de tiempo más corto (2ns) que permite la red. Los periodos de tiempo probados fueron de 50ns, 25ns, 10ns, 4ns, 2ns hasta 1ns con un ciclo de trabajo de 50/50.

El tiempo total al enviar 16 paquetes de 240 bits cada uno, por entradas diferentes a salidas diferentes es de 580ns. La latencia de cada paquete es de 40ns desde el momento en que se envía el dato a la entrada hasta la salida, donde 10ns es lo que tarda la red en hacer la asignación del canal y enviar un paquete desde su origen a su destino, al enviar los datos en una condición óptima donde la red está siempre ocupada, se aprovecha el rendimiento de envío de datos, observando a la salida entre un paquete y otro de 6ns (ver Fig. 5). Tomando los 40ns como latencia total del envío de paquetes de 240 bits, se tiene una velocidad de transmisión de 240bits/40ns = 750MB/s.

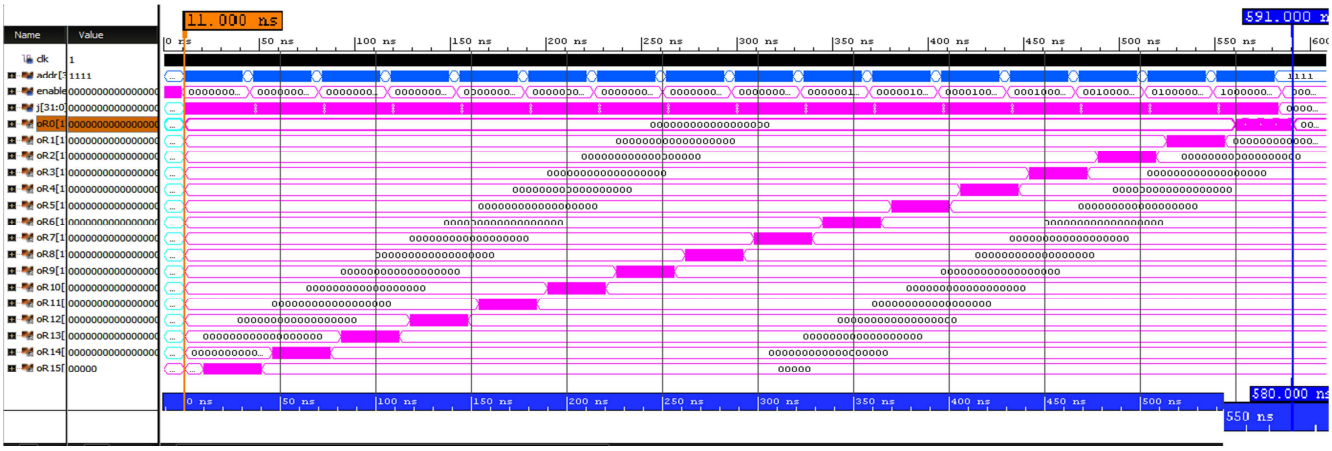

Fig. 14. Envío de 16 paquetes a 16 salidas con periodo de 2ns.

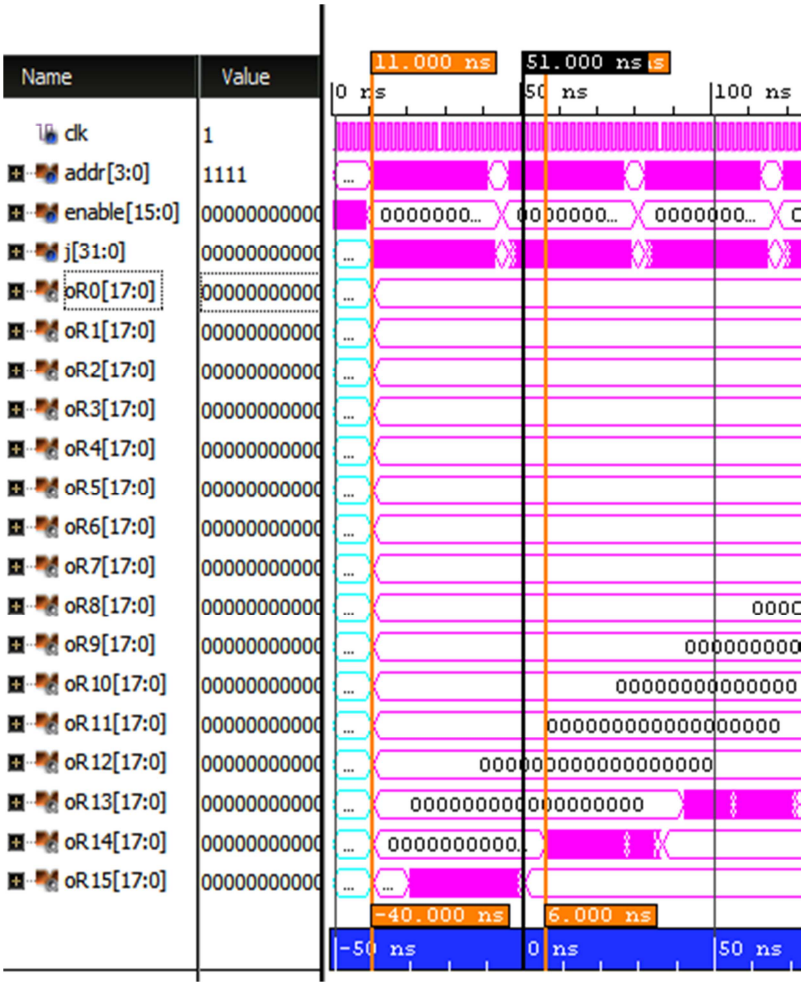

Fig. 15. Retardo de 6ns en el envío entre un paquete y otro.

En la Fig. 6 se muestra la correcta entrega de paquetes a la salida, se observa que 10ns después (línea vertical azul) de que la señal ‘addr’ (fila azul) envía “0000”, que representa la primera dirección a leer de la memoria, se recibe el flit de cabecera en la terminal-destino ‘oR0’ (fila amarilla), donde después de haber recorrido los bits de destino en dicho flit llega al destino solo con los bits que lo identifican que es un flit de cabecera, el siguiente flit contiene el identificador del flit “10” y una secuencia de 4 veces “1100”, con ello se corrobora la asignación adecuada del canal para dicho paquete de datos, y que se reciben de forma completa y sin errores.

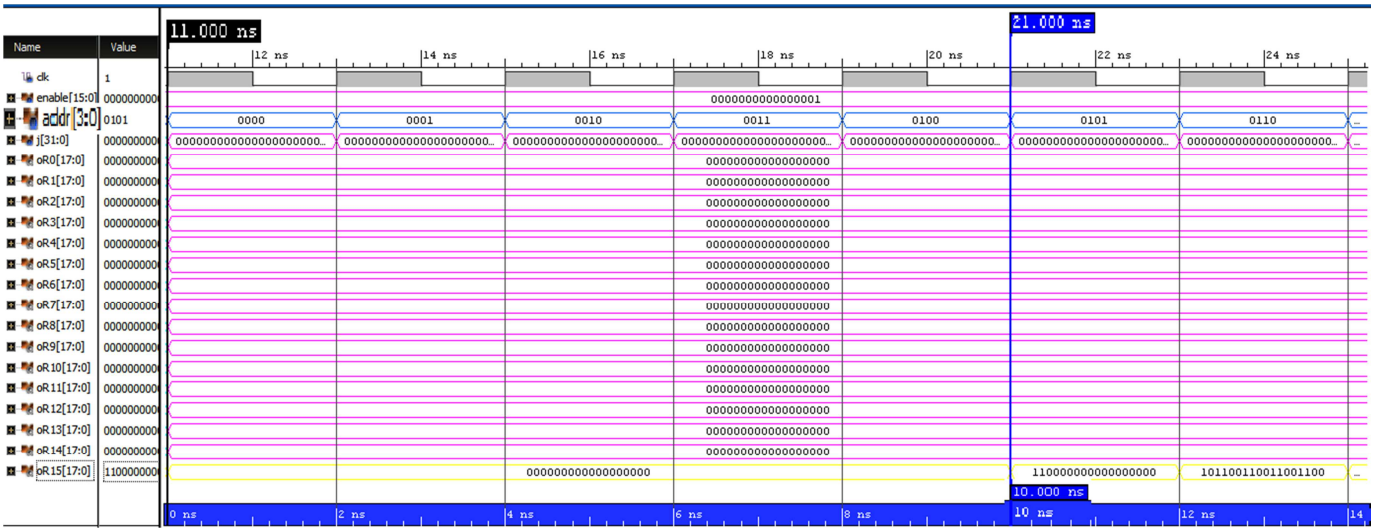

Fig. 16. Recepción correcta de los flits.

Se anexa una síntesis de los resultados de las pruebas realizadas, las que consisten en enviar los diferentes tamaños de paquetes a través de la red utilizando distintos periodos de tiempo, obteniendo en los casos de periodos de tiempo de 50ns hasta 2ns la recepción integra de la información enviada. Esta información presenta el comparativo de la latencia usando diferentes periodos de tiempo con diferentes tamaños de paquetes, se anexan los resultados en la Fig 7 y Tabla 1.

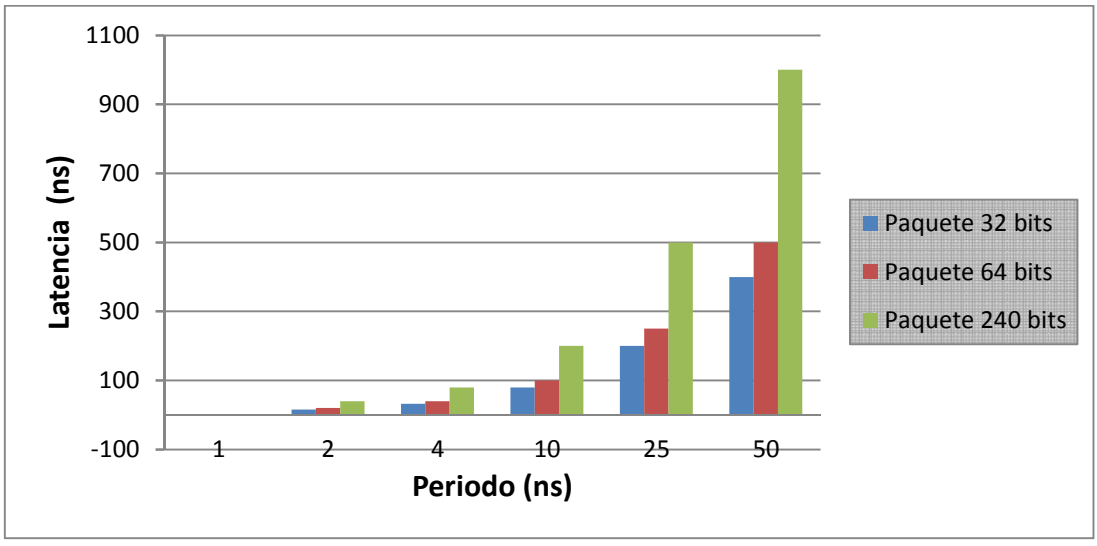

Fig. 17. Relación tamaño de los paquetes enviados contra latencia.

Periodo	Paquete (bits)	Latencia por paquete	Costo de Ocupación	Envió completo
50ns	240	1000ns	LUTs-> 816(1.53%) FF -> 18 (0.67%) BFG -> 1(3.12%)	Si
	64	500ns		Si
	32	400ns		Si
25ns	240	500ns	LUTs-> 816(1.53%) FF -> 18 (0.67%) BFG -> 1(3.12%	Si
	64	250ns		Si
	32	200ns		Si
10ns	240	200ns	LUTs-> 816(1.53%) FF -> 18 (0.67%) BFG -> 1(3.12%	Si
	64	100ns		Si
	32	80ns		Si
4ns	240	80ns	LUTs-> 816(1.53%) FF -> 18 (0.67%) BFG -> 1(3.12%)	Si
	64	40		Si
	32	32ns		Si
2ns	240	40ns	LUTs-> 816(1.53%) FF -> 18 (0.67%) BFG -> 1(3.12%)	Si
	64	20ns		Si
	32	16ns		Si
1ns	240	N/A	N/A	No
	64			No
	32			No

Tabla 1. Resultados del análisis y evaluación de la red.

-

4. Conclusiones

El diseño presentado en este trabajo tiene como resultad la arquitectura de una NoC con muy poco nivel de ocupación en el dispositivo FPGA, menor al 2%, lo que deja la mayor lógica programable a disposición del diseño de los elementos de procesamiento, los cuales pueden ser conectados a la red e intercomunicarse entre ellos hasta una tasa de transferencia de 750 MB/s. Esta arquitectura ha sido diseñada de forma genérica, esto es que no tiene elementos particulares de la tarjeta donde se ha implementado permitiendo que el diseño pueda ser exportado a cualquier dispositivo FPGA, así como facilitar la escalabilidad a sistemas con más nodos de entrada, salida, y continuar explotando las características de las FPGAs. El diseño propuesto también muestra la versatilidad del enrutamiento de paquetes, iniciando la propagación del paquete en los nodos que han liberado el canal a pesar que no haya concluido su llegada a la terminal-destino. También, es posible notar que al establecer un periodo de 1ns, no hay resultados de latencia, esto es debido a que las características del sistema implementado no soporta velocidades tan altas, para poder lograr una tasa de transferencia mayor a las ya obtenidas, una opción puede ser aumentar el tamaño de los flits que se transmiten, teniendo un impacto directo en la ocupación que el diseño tiene en la FPGA.

6. Referencias

[1] M. Lanzagorta, S. Bique, R. Rosenberg, R. O. Rosenberg, “Introduction to Reconfigurable Supercomputing”. Morgan & Claypool Publishers. Vol. 4. No. 1. Enero 2009.

[2] N. Kapre, N. Mehta, M. DeLorimier, R. Rubin, H. Barnor, M. Wilson, M. Wrighton, and A. DeHon, “Packet Switched vs. Time Multiplexed FPGA Overlay Networks”, 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. Apr. 2006. 205-216 pp.

[3] Ye Lu, J. McCanny, S. Sezer, “Generic Low-Latency NoC Router Architecture for FPGA Computing Systems”. Field Programmable Logic and Applications (FLP). 2011. 82-89 pp.

[4] W. J. Dally, “Route packets, not wires: on-chip interconnection networks”. Proceedings of the 38th Design Automation Conference IEEE. No. 01CH37232.

[5] N. Kapre, “Packet-switched on-chip FPGA overlay networks”. PhD. Dissertation. 2006. 6-30 pp.

[6] W.Dally, B.Towles, Principles and Practices of Interconection Networks. 1º edición. 2004. Morgan Kaufman Publishers. San Fransico. 13-36 pp.

[7] G. Dimitrakopoulos, A. Psarras, I. Seitanidis, Microarchitecture of Networks on Chip Routers a designer’s perspective. 1º edicion. 2015. Springer. Londres. 11-18, 61-71 pp.

[8] Zynq-700 All Programmable SoC Overview, Preliminary Product Specification. Xilinx. Estados Unidos de America. 2014.