



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y
ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, REDES Y
COMUNICACIÓN DE DATOS**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA, REDES Y COMUNICACIÓN
DE DATOS**

**TEMA: ESTUDIO DE UNA NETWORK INTERFACE PARA
UNA ARQUITECTURA NOC BASÁNDOSE EN LOS
PARÁMETROS DE DESEMPEÑO DE UNA RED**

AUTOR: BYRON FERNANDO OJEDA BÁEZ

DIRECTOR: Msc. ALULEMA DARWIN

CODIRECTOR: DR. FLORES MARCO

SANGOLQUÍ

2015

CERTIFICADO

Certificamos que el proyecto titulado “ESTUDIO DE UNA NETWORK INTERFACE PARA UNA ARQUITECTURA NOC BASÁNDOSE EN LOS PARÁMETROS DE DESEMPEÑO DE UNA RED”, realizado por Byron Fernando Ojeda Báez, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE en su reglamento.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

Atentamente,



Msc. Alulema Darwin
DIRECTOR



Dr. Flores Marco
CODIRECTOR

DECLARACIÓN DE RESPONSABILIDAD

OJEDA BÁEZ BYRON FERNANDO

DECLARO QUE:

El proyecto de grado denominado “ESTUDIO DE UNA NETWORK INTERFACE PARA UNA ARQUITECTURA NOC BASÁNDOSE EN LOS PARÁMETROS DE DESEMPEÑO DE UNA RED”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas cuyas fuentes se incorporan en las referencias bibliográficas.

Consecuentemente declaro que este trabajo es de mí autoría, en virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto en mención.

Sangolquí, 13 de julio de 2015.



Byron Fernando Ojeda Báez

AUTORIZACIÓN

Yo, Byron Fernando Ojeda Báez, autorizo a la Universidad de las Fuerzas Armadas "ESPE" la publicación, en la biblioteca virtual de la Institución del trabajo "ESTUDIO DE UNA NETWORK INTERFACE PARA UNA ARQUITECTURA NOC BASÁNDOSE EN LOS PARÁMETROS DE DESEMPEÑO DE UNA RED", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, 13 de julio de 2015



Byron Fernando Ojeda Báez

DEDICATORIA

La presente tesis dedico de manera especial, a mi querida Madre por su todo su amor brindado y apoyo incondicional durante las diferentes etapas de mi vida, en la cuales de forma sincera y desinteresada supo brindarme los consejos necesarios y acertados para triunfar de manera personal y profesional.

A mi padre por su constante apoyo buscando siempre mi crecimiento y bienestar en todos los ámbitos de mi vida, su profesionalismo es mi ejemplo a seguir.

A mis hermanas que de manera sincera han sido un gran apoyo para lograr ser la persona que soy, gracias a sus consejos y tiempo que me han dedicado en las diferentes etapas de mi vida en los buenos y malos momentos.

A mi abuelita por sus enseñanzas, que desde niño, han sido un gran apoyo para enfrentar la vida de una manera honrada, solidaria y recíproca, desde el cielo siempre está guiando el rumbo de mi vida.

A mi abuelito por su preocupación y confianza entregada desde que era un niño, sus consejos de vida han sido muy importantes.

AGRADECIMIENTO

Agradezco de manera especial a mi Dios que día a día cuida de mí y de mi familia, lo cual ha permitido culminar con éxito este logro en mi vida.

A mis padres Rosario y Fernando que de manera sincera dedicaron su tiempo a enseñarme que la vida hay que lucharla día a día para crecer como persona siempre con la bendición de Dios, eternamente agradecido con ellos por su apoyo incondicional.

A mis hermanas: Silvana por sus consejos que desde muy pequeño me impartió y todo su apoyo brindado de manera desinteresada, Lizabeth por ser un ejemplo de dedicación que ha sido clave en mi vida y Andrea para que los logros que coseche sean motivación para su vida.

A mi sobrino Jossue, el niño que cambio mi vida, quien me llena de mucha fortaleza y alegría para seguir adelante día a día.

A mis amig@s que han sido mi apoyo en las diferentes etapas de mi vida.

A mis profesores que gracias a los conocimientos brindados he podido alcanzar este logro en mi vida.

A mis demás familiares, compañeros de estudio y trabajo por su apoyo incondicional.

A Sole, una mujer muy especial en mi vida que me ayudo con el empujón final de este proyecto, un sincero agradecimiento.

ÍNDICE DE CONTENIDO

CERTIFICADO.....	i
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN.....	iii
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xii
RESUMEN.....	xiii
ABSTRACT.....	xiv
1. INTRODUCCIÓN	1
1.1. ANTECEDENTES.....	1
1.2. JUSTIFICACIÓN E IMPORTANCIA.....	1
1.3. ALCANCE DEL PROYECTO.....	3
1.4. OBJETIVOS.....	3
1.4.1. GENERAL.....	3
1.4.2. ESPECÍFICOS	3
2. MARCO TEÓRICO	4
2.1. SYSTEM ON CHIP (SOC).....	4
2.1.1. IP CORES	5
2.2. MPSOC Multiprocessor System-on-Chip.....	9
2.2.1. INTRODUCCIÓN	9
2.2.2. TIPOS DE MPSOC	10
2.2.3. HISTORIA MULTIPROCESADORES	11
2.2.4. PRIMEROS MPSOC	11

2.3.	NOC Network on Chip	13
2.3.1.	ANTECEDENTES	13
2.3.2.	ARQUITECTURA NOC	15
2.3.3.	TOPOLOGIAS	17
2.3.4.	TÉCNICAS DE CONMUTACIÓN PARA UNA RED ON CHIP.	21
2.3.5.	TÉCNICAS DE ENRUTAMIENTO PARA UNA RED ON CHIP	23
2.3.6.	PARÁMETROS DE DESEMPEÑO DE UNA RED	28
2.3.7.	REDES ON CHIP DISPONIBLES EN EL MERCADO	29
2.4.	NETWORK INTERFACE (NI)	31
2.4.1.	ARQUITECTURAS DE COMUNICACIÓN ON-CHIP	32
2.4.2.	BUS DE INTERFAZ ESTANDAR BASADO EN SOCKET	36
2.5.	QoS (Quality of Service)	38
2.5.1.	NIVELES DE QoS	40
3.	PRINCIPALES PARÁMETROS DE LA RED ON-CHIP PROPUESTA	42
3.1.	ALTO RENDIMIENTO Y BAJA LATENCIA.....	42
3.2.	PARÁMETROS DE DISEÑO DE UNA RED ON CHIP	43
3.2.1.	TOPOLOGÍA DE LA RED	44
3.2.2.	PROTOCOLO DE CONMUTACIÓN	45
3.2.3.	ALGORITMO DE ENRUTAMIENTO	46
3.2.4.	TAMAÑO FLIT	46
3.2.5.	FRECUENCIA DE RELOJ	47
3.2.6.	TRÁFICO DE LA RED	47
4.	SIMULACIÓN.....	50
4.1.	SIMULACION TRÁFICO CBR (Constant Bit Rate).....	50
4.2.	SIMULACIÓN TRÁFICO MULTIMEDIA.....	53

4.3. ANÁLISIS DE RESULTADOS	56
CONCLUSIONES Y RECOMENDACIONES	60
Conclusiones	60
Recomendaciones	61
ANEXOS	62
A. SIMULADOR NIRGAM	¡Error! Marcador no definido.
GLOSARIO DE TÉRMINOS	63
BIBLIOGRAFÍA	65

ÍNDICE DE FIGURAS

Figura 1 SOC (System on Chip)	4
Figura 2 Tipos IP CORES	8
Figura 3 Evolución del SOC al MPSOC	9
Figura 4 Estructura simple MPSOC Heterogéneo	10
Figura 5 Arquitectura Lucent Daytona	12
Figura 6 C-5 Network Processor	13
Figura 7 Interfaz de Bus Convencional	13
Figura 8 Estructura con barra cruzada	14
Figura 9 Arquitectura tradicional de buses	15
Figura 10 Arquitectura NOC	16
Figura 11 Topología red directa punto a punto	18
Figura 12 Topología red directa tipo 2D mesh	19
Figura 13 Topología Torus	19
Figura 14 Topología de redes indirectas fat-tree	20
Figura 15 Topología red irregular malla optimizada	21
Figura 16 Canales Virtuales en técnica Wormhole	22
Figura 17 Conmutación almacenamiento y reenvío	22
Figura 18 Conmutación de corte virtual	23
Figura 19 Enrutamiento XY desde el router A hasta el router B	25
Figura 20 Posibles giros en una topología malla a) NE b) ES c) WS d) NW e) WN f) SW g) SE h) EN	27
Figura 21 Giros no permitidos algoritmo odd-even columnas pares	27
Figura 22 Giros no permitidos algoritmo odd-even columnas impares	28
Figura 23 Ejemplo ST NOC	31

Figura 24 Estructura General Network Interface.....	32
Figura 25 Arquitectura AMBA (Pasricha & Dutt, 2008)	33
Figura 26 Canales read address y read data de la Arquitectura AXI	34
Figura 27 Canales write address, write data y write response de la Arquitectura AXI	34
Figura 28 Topologías punto a punto de la arquitectura Wishbone.....	36
Figura 29 Ejemplo de un sistema implementado con un bus de interfaz estándar basada en SOCKET.....	37
Figura 30 Calidad de Servicio QoS.....	39
Figura 31 Modelo de Referencia OSI.....	44
Figura 32 Proceso de un diseño de un sistema on chip	49
Figura 33 Red on chip simulada con tráfico CBR	51
Figura 34 Elementos azulejo red on-chip.....	51
Figura 35 Tamaño flit vs Rendimiento en tráfico CBR con diferentes frecuencias de operación.....	57
Figura 36 Tamaño flit vs Rendimiento en tráfico multimedia con diferentes frecuencias de operación.....	57
Figura 37 Rendimiento vs Potencia consumida en tráfico CBR.....	58
Figura 38 Rendimiento vs Potencia consumida en tráfico multimedia	58
Figura 39 Latencia vs Rendimiento en tráfico CBR	59
Figura 40 Latencia vs Rendimiento en tráfico multimedia.....	59
Figura 41 Estructura de un azulejo	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 1 Características IP Cores	7
Tabla 2 Componentes arquitectura NOC	17
Tabla 3 Arquitectura AXI y AMBA	35
Tabla 4 Parámetros QoS	39
Tabla 5 Rangos Parámetro Hurst	48
Tabla 6 Valores parámetro hurst, máxima y mínima trama de la traza de series de tv	49
Tabla 7 Parámetros de configuración simulación con tráfico CBR	50
Tabla 8 Parámetros obtenidos tráfico CBR con frecuencia de operación de 500Mhz.	52
Tabla 9 Parámetros obtenidos tráfico CBR con frecuencia de operación de 700Mhz.	52
Tabla 10 Parámetros obtenidos tráfico CBR con frecuencia de operación de 1Ghz.	53
Tabla 11 Parámetros de configuración simulación con tráfico multimedia... ..	53
Tabla 12 Parámetros obtenidos tráfico multimedia con frecuencia de operación de 500Mhz.	54
Tabla 13 Parámetros obtenidos tráfico multimedia con frecuencia de operación de 700Mhz.	55
Tabla 14 Parámetros obtenidos tráfico multimedia con frecuencia de operación de 1Ghz.	55

RESUMEN

En este proyecto se realiza un estudio del arte referente a System on Chip (SOC), Multiprocessor System-on-Chip (MPSOC), Network on Chip (NOC), y lo concerniente a Network Interface (NI) detallando el tema de los IP Cores que son bloques externos que comunican con la red on-chip. Seguidamente se realiza un análisis de los principales parámetros de desempeño de una red on chip como son: Rendimiento, latencia y potencia, así como los valores de la configuración que se debe realizar sobre la red on-chip para que el desempeño sea el más óptimo; algunos de estos valores se encuentran relacionados entre sí: algoritmo de enrutamiento, topología de la red, frecuencia de operación, entre otros; los mismos se basan en las aplicaciones más usadas en la actualidad tales como, aplicaciones en tiempo real y aplicaciones multimedia, finalmente se simulará la red on-chip propuesta sobre el software NIRGAM en su versión 2.1 para poder validar que los resultados obtenidos sean los esperados o a su vez explicar ciertas observaciones sobre los mismos, en el análisis se toma en cuenta las características de la tarjeta FPGA disponible en la Universidad de las Fuerzas Armadas ESPE (Virtex-6 ML605) para una futura implementación.

PALABRAS CLAVE

- **REDES ON-CHIP**
- **INTERFAZ DE RED**
- **PARÁMETRO HURST**
- **IP CORE**
- **MULTIPROCESSOR SYSTEM-ON-CHIP**

ABSTRACT

In this project an art studio reference is made to System on Chip (SOC), Multiprocessor System-on-Chip (MPSOC), Network on Chip (NOC), and with regard to Network Interface (NI) detailing the issue of IP Cores which are external blocks that communicate with the on-chip network. Following an analysis of the main performance parameters of a network on chip it is performed as: throughput, latency and power, as well as configuration settings to be performed on the on-chip network that is the most optimal performance ; Some of these values are interrelated: routing algorithm, network topology, operating frequency, among others; they are based on the applications most used today such as real-time applications and multimedia applications, on-chip finally proposed network on NIRGAM software version 2.1 will be simulated to validate that the results are as expected or turn explain certain observations on them, the analysis takes into account the characteristics of the FPGA board available in the University of the Armed Forces ESPE (Virtex-6 ML605) for future implementation.

1. INTRODUCCIÓN

1.1. ANTECEDENTES

Los circuitos integrados son la base fundamental del desarrollo de la electrónica en la actualidad, debido a la tendencia a facilitar y economizar las tareas del hombre, la integración de los mismos está creciendo rápidamente gracias a la utilización de nuevas técnicas y herramientas de diseño e implementación, lo que ha permitido disminuir la brecha entre la capacidad de integración de transistores dentro de un chip y el diseño de los mismos.

Para satisfacer los requerimientos del mercado actual en cuanto a carga computacional, se utilizan sistemas embebidos basados en SOC que incorporan un solo procesador o chip maestro, sin embargo, incluso estas soluciones se ven desbordadas y se requiere buscar soluciones alternativas.

En una aplicación de alto nivel computacional, se utilizaban varios SOC que deben compartir el control y procesamiento general de la aplicación creándose un problema de paralelismo, desperdicio de transistores, consumo de potencia y limitación física, de ahí que aparece la necesidad de desarrollar los Multiprocessor System-on-Chip (MPSOC). El MPSOC es un SOC compuesto por múltiples procesadores, memorias y circuitería especializada, interconectados a través de una infraestructura de comunicaciones.

1.2. JUSTIFICACIÓN E IMPORTANCIA

La comunicación en los dispositivos antiguos era simple, bastaba con tener una jerarquía de buses, comúnmente se utilizaban dos tipos de buses uno de alta disponibilidad y otro de baja disponibilidad, pero al llegar la idea de integrar varios dispositivos en un único chip (MPSOC), la jerarquía de buses presenta varios inconvenientes, que surgen a partir de la convergencia de aplicaciones, ya que la misma demanda la combinación de varios tipos de tráfico en el mismo SOC, por lo cual se necesita tratar el tema QoS dentro de la red, lo que lleva a cambiar completamente la interconexión on Chip, la misma que debe brindar una estructura flexible, reconfigurable y reprogramable, para lo cual surge un nuevo paradigma de interconexión entre los distintos módulos dentro de un mismo chip, empleando Network on chip (NOC).

La arquitectura NOC, promete ser una solución fiable en la interconexión on Chip, capaz de reemplazar con garantías una interconexión basada en buses, ya que es una arquitectura que tiene prestaciones como la escalabilidad, flexibilidad, eficiencia energética, aumento del ancho de banda. La arquitectura NOC está basada en redes de conmutación de paquetes, consta básicamente de dos bloques fundamentales que son: switches, cuya principal función es la de encaminar los paquetes de su origen a su destino; y Network interfaces que se necesitan para conectarse a cada núcleo del NOC y convertir las solicitudes de transacción/respuestas en paquetes y viceversa. Para mejorar el desempeño de la NOC se debe mejorar las prestaciones tanto de los switches como de la Network interface, pero específicamente el énfasis estará en la optimización de la network interface.

El estudio de interfaces de red (network interface) en redes de conmutación de paquetes convencionales, ha generado un alto interés y ha logrado plantear diseños muy eficientes con alto throughput y baja latencia. Sin embargo, estas características se han obtenido con el uso de procesadores dedicados a la comunicación y grandes buffers de almacenamiento.

Al hablar de sistemas embebidos, se desea un reducido consumo de potencia y una alta velocidad de transmisión utilizando una mínima capacidad de almacenamiento (buffering). Como consecuencia, pretender utilizar los interfaces de red convencionales en SOC (System-on-chip) es inviable, dado que su costo de implementación es muy alto a nivel de chip y considerando además que el tamaño de los IP-cores que se van a conectar a la NOC es relativamente pequeño.

Existen varias propuestas de diseño de interfaces de red con pequeñas áreas de implementación. Sin embargo éstas aún no proveen el throughput y la latencia deseable para estructuras complejas de SOC como es el caso de un MPSOC (Multi-Processor-System-on-chip).

A fin de mejorar la eficiencia de la red es importante estudiar, proponer, desarrollar e implementar una Network interface intra-chip con un costo de implementación bajo y características de comunicación aceptables

1.3. ALCANCE DEL PROYECTO

Realizar un estudio del estado del arte referente a System on Chip (SOC), Multiprocessor System-on-Chip (MPSOC), Network on Chip (NOC), y lo concerniente a Network Interface (NI). Luego se analizan los principales parámetros de desempeño de una red on chip para simular la misma con diferentes tipos de tráfico como son, tráfico CBR y tráfico multimedia que son los más usados en la actualidad.

1.4. OBJETIVOS

1.4.1. GENERAL

Realizar un estudio sobre los principales parámetros de desempeño de una Red on Chip, analizando las configuraciones más adecuadas que puedan realizarse sobre la misma considerando que en trabajos futuros será implementada en una tarjeta FPGA.

1.4.2. ESPECÍFICOS

- Realizar un estudio profundo del estado del arte de Network Interface usadas en conexiones on chip que se compatible con las arquitecturas de comunicaciones más comunes, así como que permita conectar varios IPCores con los routers.
- Establecer los principales parámetros de la red on chip que se deben considerar para obtener valores de rendimiento y latencia óptimos, inyectando tráfico CBR y multimedia.
- Desarrollar simulaciones del comportamiento de una red on chip utilizando para ello el programa NIRGAM con diferentes tipos de tráfico.

2. MARCO TEÓRICO

2.1. SYSTEM ON CHIP (SOC)

En la actualidad la industria de los semiconductores tiende a mejorar el alcance de los circuitos integrados de gran escala, para poder mantener el crecimiento de los niveles de integración, es por este motivo, que se han desarrollado nuevas tecnologías que soportan el aumento de la complejidad en estos circuitos integrados, una de estas tecnologías es el SOC (System on Chip).

SOC es un sistema hardware - software muy complejo que integra en un solo chip los diferentes componentes de un sistema, para poder brindar características de alto rendimiento, ya que permite construir dentro del mismo sustrato de silicio, sistemas que contengan componentes analógicos, digitales y mixtos, por ejemplo sistemas inalámbricos, sistemas de control en tiempo real, sistemas de exploración espacial, entre otros [1].

Los SOC son diseñados sobre una placa FPGA, cuya principal ventaja es que son reprogramables, lo que añade una enorme flexibilidad en el flujo del diseño [2] [1] [3]. En la Figura 1 se observa la estructura de una SOC.

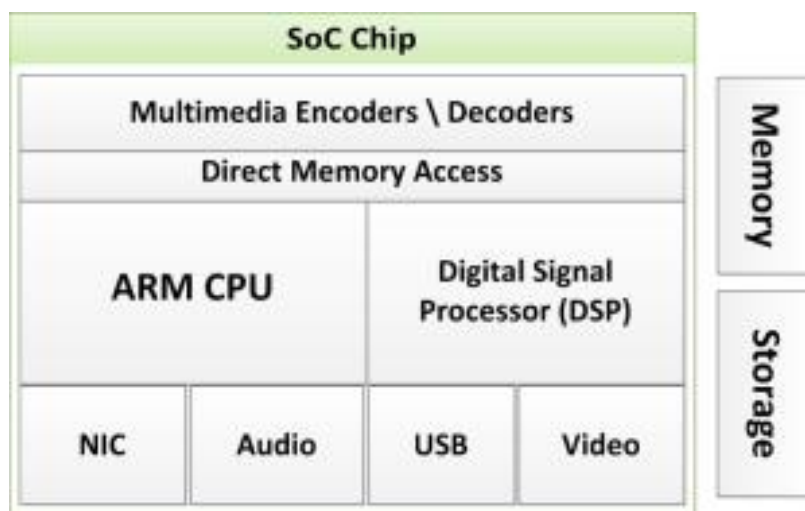


Figura 1 SOC (System on Chip) [4]

El principal requisito para la creación de diseños SOC, es el conjunto de bloques IP (Intellectual Property) reutilizables que lo componen, que ayudan a la integración plug and play.

La combinación de los diferentes bloques IP, de distintos tipos, áreas y configuraciones, son la clave para que el SOC pueda cumplir óptimamente con las necesidades de la aplicación destino [1].

2.1.1. IP CORES

Son bloques funcionales complejos previamente diseñados y verificados para diferentes niveles de implementación, son re-utilizables para la creación de distintos diseños SOC, de acuerdo a la necesidad de la aplicación. Esto brinda la ventaja de agrupar los componentes dentro de un mismo chip y disminuir notablemente el tamaño y consumo de potencia de los productos ofrecidos. Existen tres tipos de IP Cores que se describen a continuación [5].

2.1.1.1. TIPOS DE IP CORES

Los IP Cores según su implementación son:

SOFT CORES

Soft Core consiste en un conjunto de archivos HDL independientes de la tecnología. La principal característica de este tipo de IP Cores es que ofrecen la más alta flexibilidad, portabilidad a todas las tecnologías y reutilización, por otro lado lo referente al rendimiento del mismo es muy impredecible, así como la falta de previsibilidad en términos de tiempo, la seguridad de este tipo de IP Cores es débil, ya que el código fuente del mismo debe ser proporcionada, por lo que se presentan desafíos en la protección del IP Core.

La mayoría de los núcleos IP de FPGA se proporcionan como *soft core*. Ya que proporcionan la posibilidad de que los usuarios ajusten parámetros, la creación de un mayor potencial de reutilización. Los *firm core* son entregados en un formato cifrado, de modo que el RTL no es visible para el usuario, pero la colocación y enrutamiento son todavía flexibles.

FIRM CORES

Este tipo de IP Cores son cajas negras encriptadas, poseen información detallada del componente (netlist) y de la descripción RTL, el rendimiento sigue siendo impredecible. Pueden ser configurados para ciertas aplicaciones, poseen una flexibilidad tecnológica limitada pero buena predictibilidad en implementación. La seguridad de este tipo de IP Cores al momento de descifrar es mucho más fácil que la del *soft core*.

HARD CORES

Son módulos de tecnología y máscara dependiente, se tiene disponible información sobre el diseño físico, por lo cual, a diferencia de las anteriores posee un rendimiento predecible del área y rendimiento de la respuesta debido a su representación a bajo nivel; los núcleos no se pueden personalizar para un diseño en particular, la portabilidad es mínima debido a que está optimizada a una tecnología específica, y los costos de desarrollo son mayores, pero el tiempo de prueba y validaciones son menores, ya que vienen incorporado en el diseño. Este tipo son los mejores para *plug and play* de las aplicaciones, pero son menos portátiles y flexibles que los otros dos tipos, poseen una fácil protección intelectual [6] [7] [8] [9].

La clasificación de los IP Cores según su licencia de uso se clasifican en:

IP CORES PROPIETARIAS

Este tipo de IP Cores requiere de una licencia comercial para su uso legal.

IP CORES ABIERTAS (OPENCORES).-

Equivalente al software libre, existen diseños de IP cores que pueden emplearse con licencias y características similares a las de este tipo de software. [10].

Las características de los tipos de IP Cores descritos se resumen en la Tabla 1 [3], donde se visualiza una comparación de cada uno de ellos referente a los niveles de abstracción, descripción, portabilidad, flexibilidad, previsibilidad y la protección que tienen para los IP Cores.

Tabla 1 Características IP Cores

TIPO	SOFT CORE	FIRM CORE	HARD CORE
NIVEL DE ABSTRACCION	RTL, gate level	Gate level, layout	Layout
DESCRIPCION	VHDL, Verilog	Netlist	Descripción de transistores
PORTABILIDAD	A todas las tecnologías	Limitada a tecnologías probadas	Optimizada a una tecnología específica
FLEXIBILIDAD	Alta	Limitada	Muy poca
PREVISIBILIDAD	Baja	Buena	Alta y definida por la tecnología
PROTECCION PROPIEDAD INTELECTUAL	Difícil	Fácil	Fácil

[3]

En la Figura 2 se puede apreciar la relación entre previsibilidad, desarrollo y complejidad versus la flexibilidad, esto para cada uno de los IP CORES.

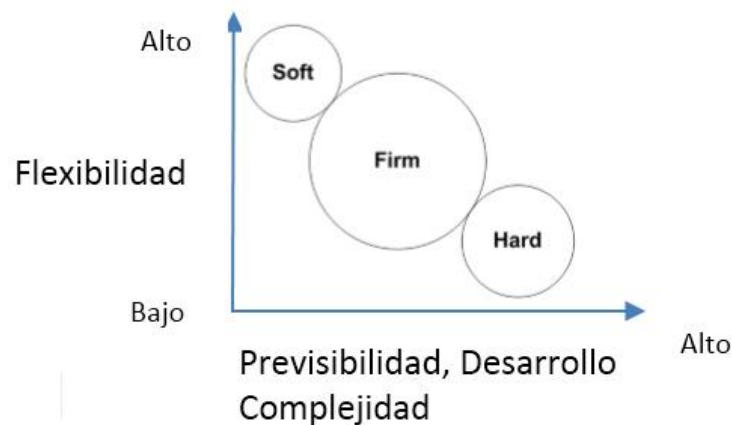


Figura 2 Tipos IP CORES

Existen organizaciones que promueven las normas de reutilización de diseño, cuyo objetivo principal es facilitar el uso de los IP Cores en los diseños, lo cual conlleva a simplificar el diseño de la interfaz de los IP Cores externos, a continuación se describen los grupos interesados en este tema.

VSIA (Virtual Socket Interface Alliance)

Esta alianza nace a partir de la necesidad de crear un nuevo conjunto de estándares para la industria del diseño de sistemas, fue creada en el año de 1996 con la visión de acelerar el desarrollo del SOC, especificando estándares abiertos que ayuden a “mezclar y combinar” IP Cores de múltiples fuentes.

La filosofía que maneja es que si se puede mezclar de una manera rápida componentes físicos en una placa de circuito impreso, los IP Cores de una forma estandarizada VC (Componente Virtual), también pueden ser fáciles de mezclar dentro de un SOC.

OPENMore

Mentor Graphics en 1999 lanzó la herramienta OPENMore, que permite evaluar la reutilización de los soft and hard Cores para diseños SOC.

Está basado en el Manual Reutilización Metodología para diseños System-on-a-Chip (RMM), el cual describe un conjunto de mejoras para la creación de diseños reutilizables, para poder usarlos en una metodología de diseño SOC. Estas mejoras están basadas en la experiencia de los autores, en el desarrollo de diseños reutilizables, así como la experiencia de los equipos de diseño en muchas empresas de todo el mundo. Tecnologías de silicio y de herramientas se mueven tan rápido que muchos de los detalles de

diseño para la reutilización indudablemente continuarán evolucionando con el tiempo. Sin embargo, los aspectos fundamentales de la metodología descrita en este libro se han convertido ampliamente adoptados y es probable que constituyan la base de diseño de chips [8] [11].

2.2. MPSOC Multiprocessor System-on-Chip

2.2.1. INTRODUCCIÓN

La empírica ley de Moore no solo describe que cada dos años se duplica el número de transistores en un circuito integrado, sino que a su vez impone nuevos retos de diseño que satisfagan los requerimientos de las aplicaciones actuales, que se ejecuten en tiempo real, de bajo consumo y multitarea entre otros.

Para poder lograr esto los SOC tuvieron que evolucionar a lo largo de veinte años aproximadamente para poder llegar al MPSOC en donde fue necesario integrar múltiples procesadores en un solo chip.

En la

Figura 3 Evolución del SOC al MPSOC

se observa la evolución del SOC al MPSOC.

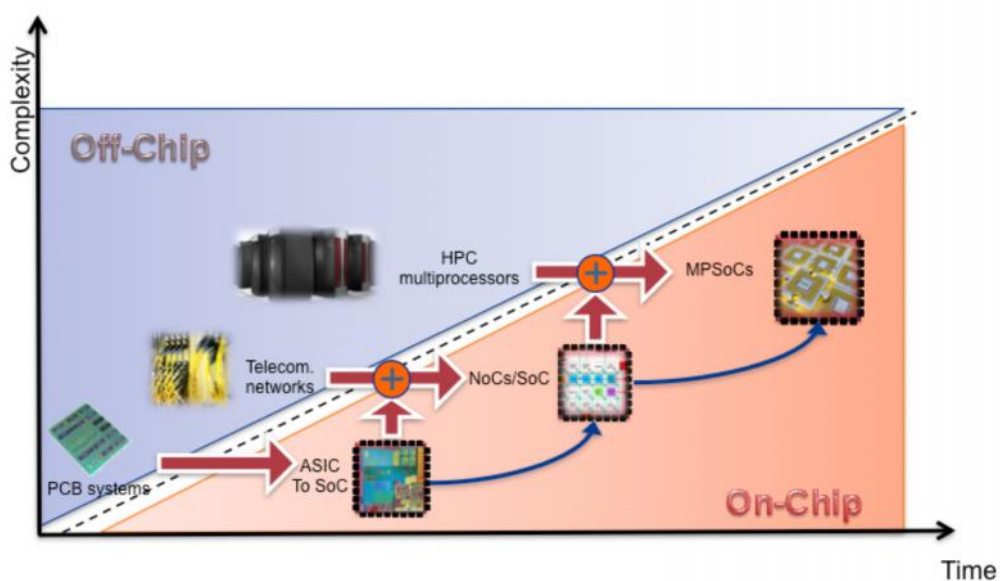


Figura 3 Evolución del SOC al MPSOC [12]

2.2.2. TIPOS DE MPSOC

MPSOC HOMOGÉNEOS

Este tipo de MPSOC proporciona hoy en día las soluciones más óptimas referentes a la eficiencia del rendimiento, son la elección más común en sistemas embebidos, posee la mejor flexibilidad y escalabilidad, pero su eficiencia en el consumo de energía no es buena, ya que son usados para sistemas de propósito general, es decir, usan un solo tipo de procesadores interconectados.

Comúnmente son usados en consolas de videojuegos, computadoras, servidores, supercomputadoras, estaciones base inalámbricas, es decir son usados para sistemas de datos en paralelo. [12] [13]

MPSOC HETEROGÉNEOS

Este tipo de MPSOC está compuesto por PEs (Processing Elements) de diferente tipo, tales como procesadores, memorias, aceleradores y periféricos.

En la Figura 4 se visualiza de forma general un MPSOC heterogéneo genérico, que está compuesto de una serie de procesadores de propósito general, algunos aceleradores (audio, video), elementos de memoria, periféricos y la infraestructura de comunicación [12].

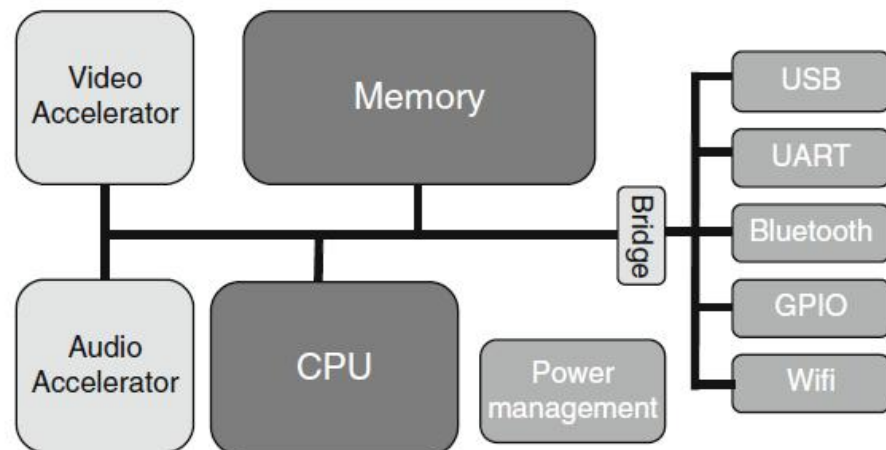


Figura 4 Estructura simple MPSOC Heterogéneo [12]

Este tipo de MPSOC provee las mejores soluciones, con respecto a la eficiencia de energía, pero tienen falencias en la escalabilidad y flexibilidad. Su programación es mucho más compleja, pero es ideal para aplicaciones en

tiempo real, ya que los conflictos entre el procesamiento de elementos y tareas es mínimo.

2.2.3. HISTORIA MULTIPROCESADORES

Los MPSOC inician su historia a partir de los multiprocesadores, cuyo concepto se reduce al de “trabajo en equipo”, es decir dividir una tarea específica en varias unidades de ejecución.

En los años 80 y principios de los 90 un microprocesador no podía caber en un chip, ya que naturalmente necesitaban múltiples chips y procesadores, a fines de los 90 la velocidad de los sistemas monoprocesadores crece mucho más rápido que la velocidad del sistema paralelo, además se logra ajustar un microprocesador en un solo chip.

La ley de Moore es identificada como una gran influencia en el desarrollo del microprocesador, la misma expresa que aproximadamente cada dos años se duplica el número de transistores en un circuito integrado hasta el 2010 el número de transistores era de 10.000.000.

2.2.4. PRIMEROS MPSOC

En la década de los 90 se llevó a cabo el desarrollo de varios uniprocesadores VLSI, ya que se crearon aplicaciones embebidas para las comunicaciones y lo referente a multimedia. A su vez el nivel de paralelismo aumento con la creación de chips con arquitectura VLIW. [13]

LUCENT DAYTONA

El primer MPSOC que se conoce fue Lucent Daytona, introducido en el año 2000, en donde se consiguió la integración de múltiples matrices de procesadores homogéneos, la arquitectura del mismo se observa en la Figura 5.

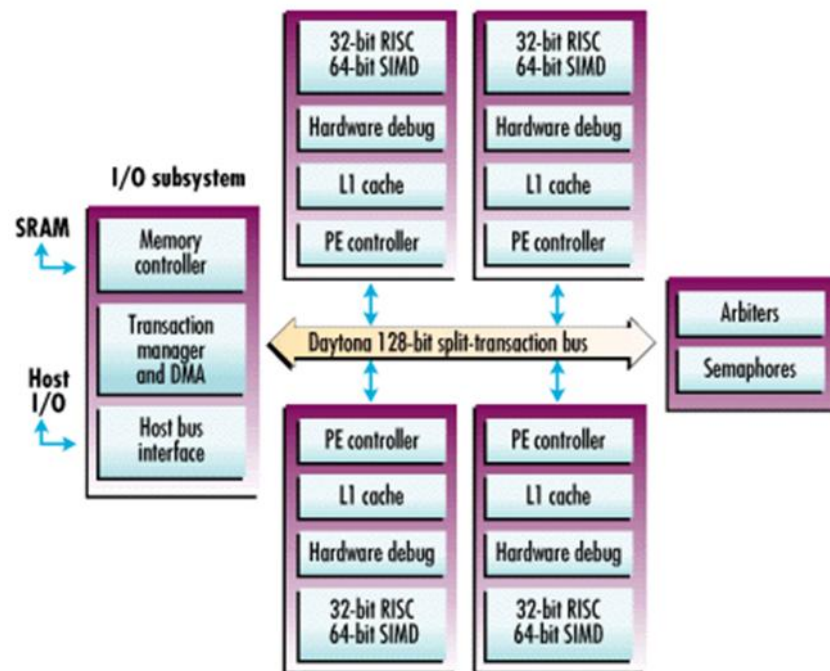


Figura 5 Arquitectura Lucent Daytona [14]

Daytona fue desarrollado para elaborar algoritmos de procesamiento de señal típicos de las estaciones inalámbricas base, en donde el procesamiento de la señal es idéntica en los cuatro canales de datos, este es un caso perfecto de arquitectura paralela.

A partir de Daytona en el año 2000, varios MPSOC se dieron a conocer, los mismos que variaron su arquitectura ya que soportaban aplicaciones que incluían video, multimedia, juegos, telefonía móvil entre otros. [13] [14].

C-5 NETWORK PROCESSOR

Haciendo referencia a las aplicaciones inalámbricas se encuentra con el MPSOC C-5 Network Processor, que a diferencia de Lucent Daytona, es un MPSOC Heterogéneo ya que se encuentra compuesto por diferentes PEs (Processing Elements) tales como procesadores, memorias, aceleradores y periféricos, funcionan a una frecuencia de 166Mhz- 233Mhz.

Fue diseñado para el procesamiento de paquetes en una red, los paquetes se manejan por 16 procesadores, agrupados en 4 grupos de 4 procesadores cada uno

Es el primer miembro de la familia C-Port™ de procesadores de red, su alto nivel de rendimiento y características que brinda, lo han convertido en la base fundamental para el desarrollo de productos y servicios que hacen

referencia a los procesadores de red, la Figura 6 se muestra la arquitectura de este procesador [13]

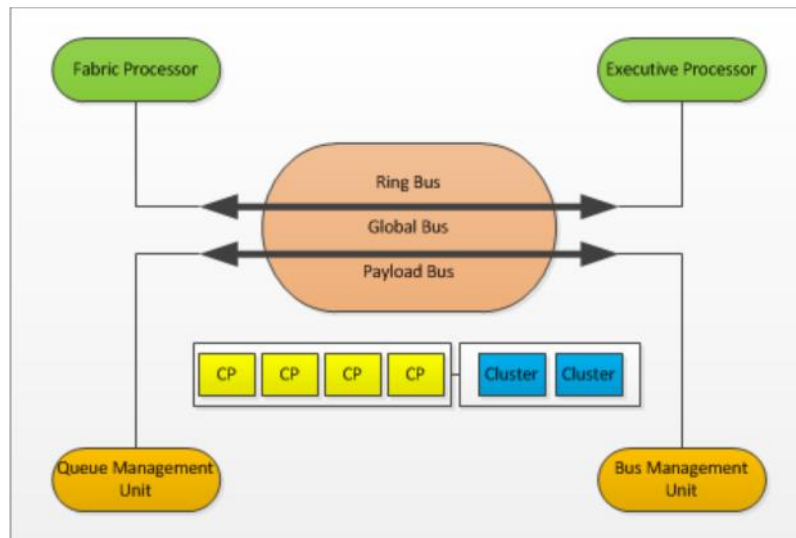


Figura 6 C-5 Network Processor [13]

2.3. NOC Network on Chip

2.3.1. ANTECEDENTES

La próxima generación de los MPSOC, como se prevé contendrá cientos o miles de núcleos, por lo cual para poder transferir datos entre los núcleos en el chip, se requiere una interconexión de alto rendimiento.

La primera generación de interconexiones en el chip fue mediante una interfaz de bus convencional. Básicamente en un bus, la interconexión en su mayoría son solo cables que interconectan los IPCores tal y como se muestra en la Figura 7 [15].

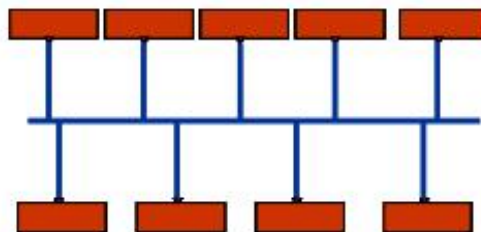


Figura 7 Interfaz de Bus Convencional [15]

Actualmente la interconexión mediante bus presenta muchas limitaciones, debido a que presentan problemas de latencia cuando existe una alta prioridad de accesos, que se estancan por el número de transacciones que están en curso en el bus.

El ancho de banda es un inconveniente que está limitado por la frecuencia del reloj, que va a la par con parámetros de diseño físico, tales como la longitud del cable, las mismas que son muy difíciles de superar.

Por este motivo los diseñadores empezaron a implementar estructuras con barras cruzadas tal y como se muestra en la Figura 8, con lo cual se mejora la previsibilidad y latencia, aumentan significativamente el ancho de banda, pero a un costo que no se puede despreciar, que es el aumento del número de cables. [15].

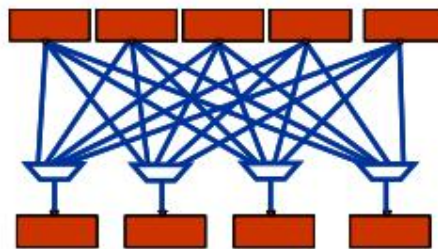


Figura 8 Estructura con barra cruzada [15]

Los diseñadores de SOC y MPSOC enfrentan nuevos retos, referente al diseño de la interconexión entre chips, que va más allá del aumento de IP Cores y requisitos de desempeño, ya que incluye la administración de energía, los múltiples dominios de reloj que puedan presentarse, requisitos de seguridad, control de errores entre otros, cuya implementación física a través de layout, se hace cada vez más difícil, ya que se presentan inconvenientes referentes al cableado, frecuencia, dominio del reloj y tensión.

En la arquitectura tradicional de buses, ver Figura, las interconexiones están determinadas por conexiones punto a punto, con un hilo dedicado a cada señal. Para grandes diseños, en particular, tiene varias limitaciones desde un punto de vista del diseño físico.

Los cables ocupan gran parte del área del chip, y en tecnología CMOS se ocupa un área dimensionada en nanómetros; las interconexiones dominan tanto el rendimiento y la disipación de potencia dinámica, como la propagación de la señal en cables a través del chip requiere múltiples ciclos de reloj [16].

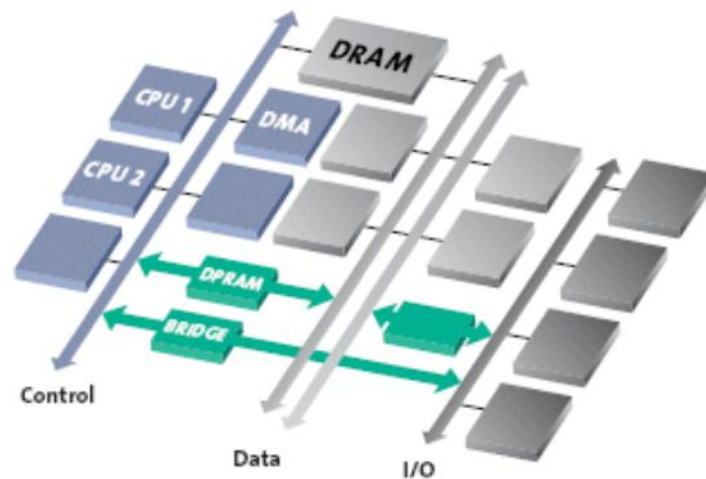


Figura 9 Arquitectura tradicional de buses [16]

La arquitectura de buses presenta varios inconvenientes, que surgen a partir de la convergencia de aplicaciones, ya que requiere la combinación de varios tipos de tráfico en el mismo SOC, por lo cual se necesita tratar el tema QoS dentro de la red, lo que lleva a cambiar completamente la interconexión on Chip, la misma que debe brindar una estructura flexible, reconfigurable y reprogramable, para lo cual surge un nuevo paradigma de interconexión entre los distintos módulos dentro de un mismo chip, empleando la arquitectura Network on chip (NOC).

2.3.2. ARQUITECTURA NOC

La arquitectura de una NOC está basada en las redes de conmutación de paquetes, y puede ser descrita por la estrategia que tiene para el enrutamiento, el control de flujo, conmutación, latencia y throughput.

La arquitectura NOC proporciona la infraestructura de comunicación para los elementos de procesamiento que componen el módulo, por lo cual deben cumplirse dos principales objetivos.

- Desarrollar el hardware de los elementos de procesamiento de forma independiente, para posteriormente añadirlos a la red, y que formen parte de la NOC.
- Dar la suficiente flexibilidad para poder escalar y configurar la red, logrando adaptar la red a las diferentes necesidades que se presenten, es decir, cuando la complejidad aumente, es decir el tráfico de la NOC sea heterogéneo [17].

La arquitectura NOC está conformada por tres principales componentes:

- Network Interface (NI)
- Switches (routers)
- Links

En la Figura 10 se visualiza una arquitectura NOC tipo mesh, compuesta de 16 Cores con sus respectivos links que interconectan los routers, la NI (Network Interface) de la arquitectura, permiten la comunicación entre un Core y un router.

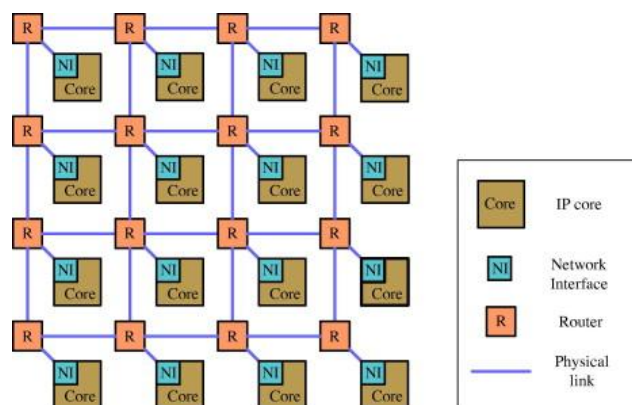


Figura 10 Arquitectura NOC [18]

La columna vertebral del NOC son los switches, cuya función principal es la encaminar los paquetes desde la fuente hacia el destino que se requiera.

En la Tabla 2 se detalla características principales de cada uno de los componentes de la arquitectura NOC.

Tabla 2 Componentes arquitectura NOC

COMPONENTE	CARACTERISTICAS
SWITCHES	Encargados de llevar a cabo la tarea de envío de paquetes dentro de la red, que va a depender del esquema de enrutamiento que se esté utilizando, generalmente tienen buffers internos, en donde se almacena información, con el fin de reducir el riesgo de pérdida de datos, debido a problemas de congestión.
LINKS	Encargados de conectar los Switches con las Network Interface, o a su vez con otros switches, pueden ser insensibles a la latencia en la red, dependiendo de la aplicación, pueden tener buffers.
NETWORK INTERFACE	Se encarga de conectar todos los IP Cores con la red, es decir, mapea las transiciones tipo bus provenientes de los IP Cores, en paquetes que puedan ser propagados dentro de la NOC así como en el lado opuesto, adicionalmente, construye las transacciones necesarias para que los paquetes puedan salir de la NOC.

[19] [20]

2.3.3. TOPOLOGIAS

La NOC puede ser caracterizada por la estructura de las conexiones de los routers, esta estructura es llamada topología.

También se puede definir a la topología como la organización de las interconexiones de la red. Es decir define como los nodos, switches, y links están conectados entre sí, las topologías pueden ser clasificadas en tres amplias categorías. Redes directas, redes indirectas y redes irregulares.

2.3.3.1. Topología Redes Directas

En este tipo de topologías, cada nodo tiene enlaces directos punto a punto a un subconjunto de otros nodos en el sistema denominados nodos vecinos.

Los nodos consisten en bloques computacionales y/o memorias, así como un bloque de la interfaz de red que actúa como un router, este router está conectado a los routers de los nodos vecinos a través de enlaces. Una de las

propiedades que hay que tener muy en cuenta en este tipo de topologías, es que si el número de nodos en el sistema aumenta, el ancho de banda total disponible en la comunicación también se incrementa.

Una mayor conectividad resulta de un mayor rendimiento, pero demanda mayores costos de energía y zona para implementaciones router y enlace.

En la Figura 11 se muestra una topología de red directa completamente conectada, en donde cada nodo está conectado directamente a todos los demás nodos, lo cual es muy prohibitivo.

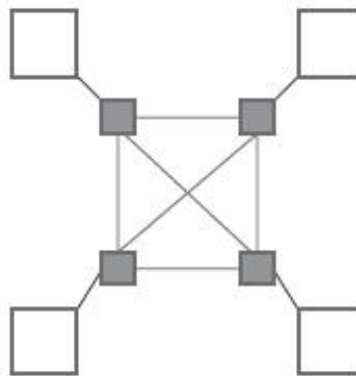


Figura 11 Topología red directa punto a punto [21]

La mayoría de las topologías de redes directas tienen una implementación ortogonal, donde los nodos pueden ser organizados en un espacio ortogonal de n dimensiones, de tal manera que cada enlace genera un desplazamiento en una sola dirección. El enrutamiento para este tipo de redes es simple, puede ser implementado en hardware, los ejemplos más comunes de este tipo de topologías son: malla n -dimensional, toro, toro cruzado, hiper cubo, y topologías octágono, en la Figura 12 se muestra un red directa malla 2-D o también denominada rejilla, la cual es una de las más populares topologías NOC, todos los enlaces tienen la misma longitud, lo cual facilita el diseño físico.

Cada nodo está conectado a cuatro de sus nodos vecinos a excepción de los nodos de borde, el área crece linealmente con el número de nodos.

Esta topología tiene la ventaja que la conexión al nodo y el algoritmo de enrutamiento es simple, lo cual facilita la implementación física.

La desventaja de esta topología es que el ancho de banda proporcionado no es alto debido a que el diámetro de la red es muy grande, así como la escalabilidad es muy pobre [22].

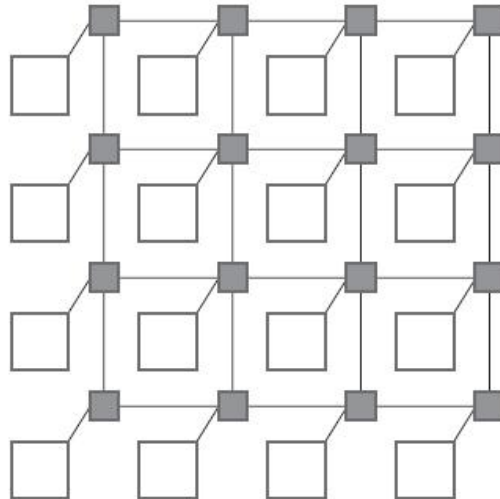


Figura 12 Topología red directa tipo 2D mesh [21]

Existe una topología denominada 2D- Torus, que básicamente tiene el mismo funcionamiento de la topología 2D Mesh, con la diferencia que en la topología Torus los nodos forman un cíclico en la 2D mesh regular, es decir, todos los routers tienen cuatro conexiones, por lo tanto, se puede decir que la topología Torus es una topología mesh con rotación en los bordes [23], [22].

La topología 2D-Torus es ideal cuando se trata de una red escalable, el throughput que maneja es mucho mejor. Un ejemplo de topología 2D- Torus se puede observar en la Figura 13.

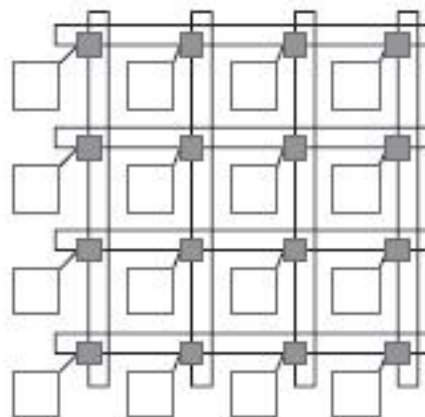


Figura 13 Topología Torus [21]

2.3.3.2. Topología Redes Indirectas

En la topología de redes indirectas cada nodo está conectado a un switch externo y cada switch tiene enlaces punto a punto a otros switches.

La interfaz de red asociada a cada nodo se conecta al puerto de un switch, estos switches no realizan ningún procesamiento de la información, así como los nodos correspondientes no realizan ninguna conmutación de paquetes.

En la Figura 14 se muestra un ejemplo de una topología de redes indirectas, específicamente una topología *fat-tree*, en este tipo de topologías los nodos esta conectados solamente a las hojas del árbol, se puede observar que para una topología de árbol sencilla la raíz y sus vecinos tienen mayor tráfico.

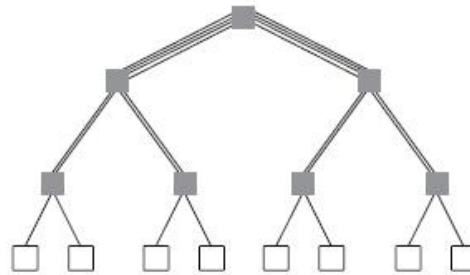


Figura 14 Topología de redes indirectas fat-tree [21]

2.3.3.3. Topología redes irregulares

Redes de topología irregulares o ad hoc son usualmente una mezcla de topologías de buses compartidos, redes directas e indirectas, el objetivo principal de este tipo de topologías es aumentar el ancho de banda en comparación con los tradicionales buses compartidos y reducir la distancia entre nodos en comparación con las redes directas e indirectas.

Las topologías de redes irregulares son personalizadas para una aplicación, un ejemplo de un topología de red irregular, se visualiza en la Figura 15, la misma es una topología malla reducida, en donde los routers y enlaces innecesarios han sido removidos.

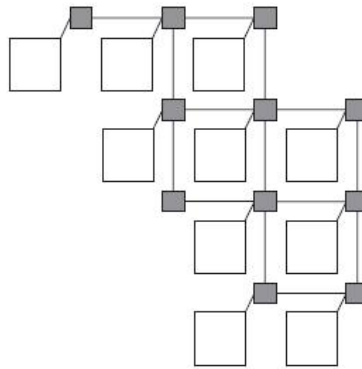


Figura 15 Topología red irregular malla optimizada [21]

2.3.4. TÉCNICAS DE CONMUTACIÓN PARA UNA RED ON CHIP

Las técnicas de conmutación están basadas en las necesidades de la red, como se mencionó con anterioridad la arquitectura NOC está basada en conmutación de paquetes. En donde la transmisión de paquetes se realiza sin utilizar reservas de todo el camino. [24]

Existe una clasificación con respecto a las redes basadas en conmutación de paquetes:

2.3.4.1. Redes Wormhole

En esta técnica, los requerimientos del buffer son reducidos a un flit, en lugar de un paquete completo.

Un flit de un paquete es enviado al router receptor, si el espacio para él es suficiente se almacena el paquete entero, caso contrario parte del paquete se distribuye en dos o más routers, si se realiza la distribución entre dos o más routers se puede producir un bloqueo de paquetes.

Para mitigar este problema, esta técnica ha adoptado canales virtuales para hacer uso eficiente de los canales físicos

Varios carriles paralelos comparten un canal virtual, como se muestra en la siguiente figura.

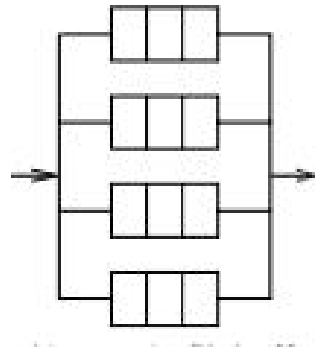


Figura 16 Canales Virtuales en técnica Wormhole [25]

La mayoría de las arquitecturas NOC utilizan esta técnica. [21], [25]

2.3.4.2. Redes de Almacenamiento y Reenvío (SAF)

Es la técnica más simple de conmutación de paquetes, un paquete es enviado desde un router al siguiente solo si el router receptor tiene espacio en el buffer para todo el paquete, por lo cual la transmisión de paquetes no se puede estancar.

Los routers envían un paquete solo cuando se han recibido en su totalidad, lo cual genera que la latencia por router y el tamaño del router sean por lo menos del tamaño del paquete, por esta restricción pocas redes on chip utilizan esta técnica [21], en la Figura 17 se puede observar este tipo de conmutación.

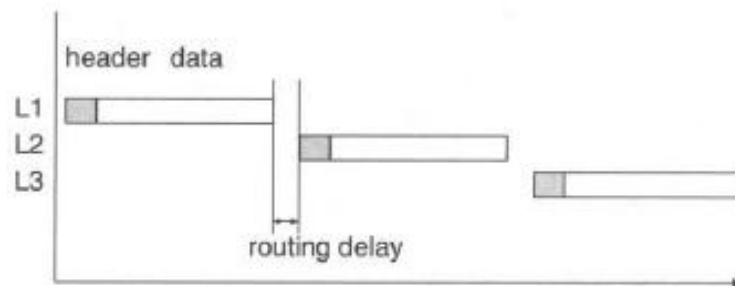


Figura 17 Conmutación almacenamiento y reenvío [26]

2.3.4.3. Conmutación de corte virtual (VCT)

Esta técnica reduce la latencia por enrutador, mediante el envío del primer flit de un paquete tan pronto como exista espacio en el siguiente router para todo el paquete, por lo tanto los demás flits siguen sin demora, caso contrario todo el paquete se encuentra almacenado. En esta técnica se reduce el tiempo

en que se almacena el paquete, la desventaja radica en el costo de implementación [27], la Figura 18 muestra este tipo de conmutación.

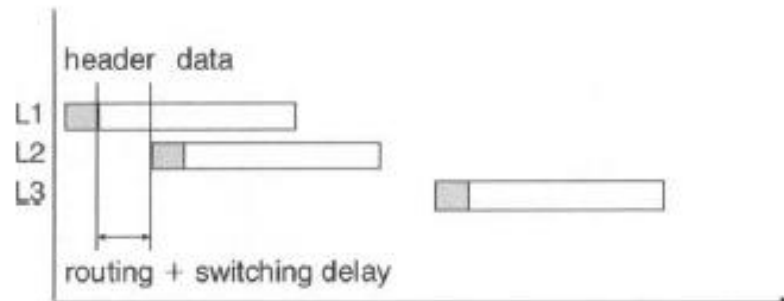


Figura 18 Conmutación de corte virtual [26]

2.3.5. TÉCNICAS DE ENRUTAMIENTO PARA UNA RED ON CHIP

El enrutamiento en una red on chip es bastante similar a la de cualquier red, el algoritmo de enrutamiento hace referencia a la forma en que los datos se enrutan del emisor al receptor, es decir es el responsable de la correcta y eficiente forma de enrutar los paquetes de la fuente al destino.

Para poder elegir un algoritmo de enrutamiento hay que tomar en cuenta ciertos parámetros tales como: reducir la potencia requerida para el enrutamiento, lo cual se logra minimizando la lógica y las tablas de enrutamiento, aumentar el rendimiento mediante la reducción de retardo y maximizando la utilización del tráfico en la red, y finalmente mejorar la robustez para adaptarse mejor a las necesidades que tenga a los cambios de tráfico. [21], [28].

Los algoritmos de enrutamiento están divididos en dos grupos:

2.3.5.1. ALGORITMOS INCONSISTENTES

Este tipo de algoritmos no disponen de información sobre las condiciones de la red, como cantidad de tráfico o congestiones que se generan.

Este tipo de algoritmos se encuentran divididos en dos sub-grupos:

2.3.5.1.1. Algoritmos Determinísticos

Este tipo de algoritmos son los más simples, ya que envían todos los paquetes desde la fuente “x” hasta el destino “y” sobre exactamente la misma ruta.

Son utilizados en redes regulares e irregulares, en redes libres de congestión, este tipo de algoritmos son fiables y brindar una baja latencia.

Tienen gran adaptación a los sistemas de tiempo real porque los paquetes siempre llegan a su destino en el orden correcto, por lo cual un reordenamiento no es necesario.

En el caso más simple cada router posee una tabla de enrutamiento que incluye rutas a todos los demás routers de la red, si la estructura de la red cambia, cada router debe actualizarse.

Este tipo de algoritmos son capaces de encontrar el próximo salto en un corto tiempo debido a que el cálculo que realizan es mínimo. [28], [29].

2.3.5.1.2. Algoritmo origen

En este algoritmo el remitente toma las decisiones acerca del enrutamiento del paquete antes de inyectar cualquier tráfico a la red.

La información acerca de todo el camino desde el origen hasta el destino está pre calculado y proporcionado en la cabecera del paquete, la misma que contiene solo la dirección destino y la ruta se calcula de forma dinámica por la participación de los routers en el camino.

Este tipo de algoritmo de enrutamiento no es aconsejable en redes que cambian constantemente de topología o tamaño.

El tamaño del paquete en este tipo de algoritmo aumenta debido a que en el paquete debe estar incluida la información de enrutamiento. [30], [28].

2.3.5.1.3. Algoritmos por la dimensión del pedido

Es un típico algoritmo de giro mínimo, el mismo determina en qué dirección se encaminan los paquetes durante todas las etapas del enrutamiento, este tipo de algoritmo es el más popular de los algoritmos determinísticos.

Está basado en la idea de que algunas topologías pueden descomponerse en varias dimensiones ortogonales es decir hiper cubo, mallas y Torus.

2.3.5.1.4. Algoritmo xy

En topología malla 2D este algoritmo es llamado XY en función de la dimensión en la que un paquete viaja primero.

Este algoritmo enruta primero los paquetes en la dirección X (dirección horizontal) a la columna correcta y luego en la dirección Y (dirección vertical) al receptor, en este algoritmo las direcciones de los routers son sus coordenadas XY .

La principal ventaja de este algoritmo es que nunca se queda en punto muerto o livelock. Una desventaja de este tipo de algoritmo es que el mismo hace que la carga sea más grande en el centro de la red, es decir el tráfico no se distribuye de manera uniforme, por lo cual es necesario incluir algoritmos para igualar la carga de tráfico en toda la red, un ejemplo de este tipo de enrutamiento se visualiza en la Figura 19.

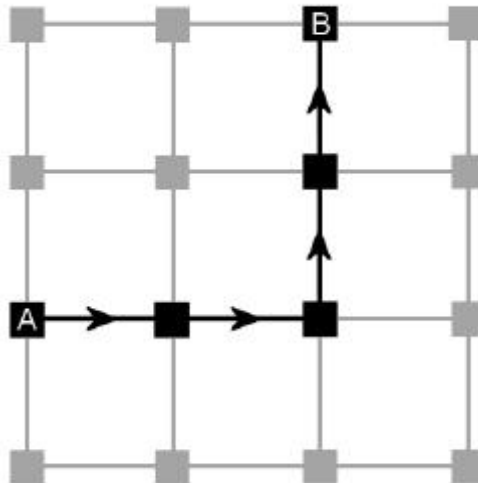


Figura 19 Enrutamiento XY desde el router A hasta el router B [28]

2.3.5.1.5. Algoritmos Estocásticos

Este tipo de algoritmos está basado en la coincidencia y la suposición de que cada paquete tarde o temprano llega a su destino. Generalmente estos algoritmos son simples y de alta disponibilidad.

El rendimiento de los datos es buena, pero presenta un inconveniente, este tipo de algoritmos son bastante lentos y utilizan muchos recursos de la red.

Estos algoritmos determinan el tiempo de vida de los paquetes (TTL), en este tiempo el paquete puede moverse dentro de la red, una vez alcanzado este tiempo el paquete es retirado de la red [28].

2.3.5.2. ALGORITMOS ADAPTATIVOS

Este tipo de algoritmos no restringe a un mensaje a un único camino cuando viaja desde la fuente hacia el destino, la decisión del camino a tomar lo hace basándose en condiciones actuales de la red, lo cual hace que el enrutamiento realizado sea mucho más flexible, reduciendo los tiempos de espera innecesarios, proporcionando una mayor tolerancia a fallos.

Este tipo de algoritmos contienen dos funciones: el enrutamiento y la selección.

- **Función de enrutamiento:** encargada de brindar un conjunto de canales de salida basados en el nodo actual y el nodo destino
- **Función de selección:** encargada de seleccionar del conjunto de canales brindados por la función de enrutamiento el más adecuado.

La función de selección suministra siempre con un canal libre, por lo cual los mensajes pueden seguir los caminos alternativos en lugar de esperar a un canal ocupado.

Los algoritmos adaptativos permiten la técnica de retroceso, la misma que hace a la cabecera dar marcha atrás en la liberación de los canales previamente reservados, buscando de este modo sistemáticamente una ruta adecuada. Para algoritmos deterministas técnica de retroceso es inútil ya que el mensaje va a ir por el mismo camino de nuevo. Los algoritmos adaptativos aumentan la flexibilidad de enrutamiento pero el hardware se vuelve más complejo y más lento. [31].

2.3.5.2.1. Algoritmo Odd-Even

Es un algoritmo adaptativo utilizado en redes on-chip dinámicas adaptativas y redes on-chip determinísticas.

El algoritmo está basado en el modelo de giro par-impar.

En una topología malla de dos dimensiones con dimensiones XY, cada nodo es identificado por su coordenada (x,y).

Un giro que se produzca en esta topología implica un cambio de 90 grados de la dirección de desplazamiento, existen ocho tipos de giros de acuerdo de acuerdo con la dirección de desplazamiento de los canales asociados, los

mismos son: ES, WS, WN, SE, SW, NE, ES y NW en donde E,W,S,N indican Este, Oeste, Sur, Norte, por ejemplo ES hace referencia si el giro se trata de un cambio de dirección de Este a Sur, en la Figura 20 se muestran los ocho giros descritos.

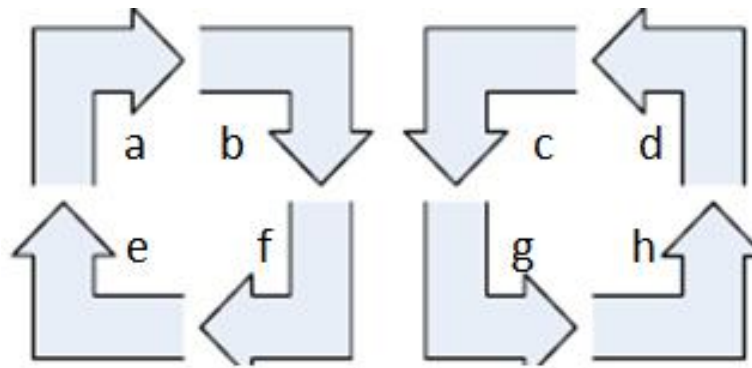


Figura 20 Posibles giros en una topología malla a) NE b) ES c) WS d) NW e) WN f) SW g) SE h) EN [30]

Existen dos teoremas en este algoritmo, que ayuda que el mismo esté libre de *deadlock*.

- Está prohibido realizar los siguientes giros en las columnas pares, se muestra en la Figura 21

EN en un nodo que se encuentra en una columna par.

ES en un nodo que se encuentra en una columna par.

- Está prohibido realizar los siguientes giros en las columnas impares, se muestra en la Figura 22

SW en un nodo que se encuentra en una columna impar.

SW en un nodo que se encuentra en una columna impar.

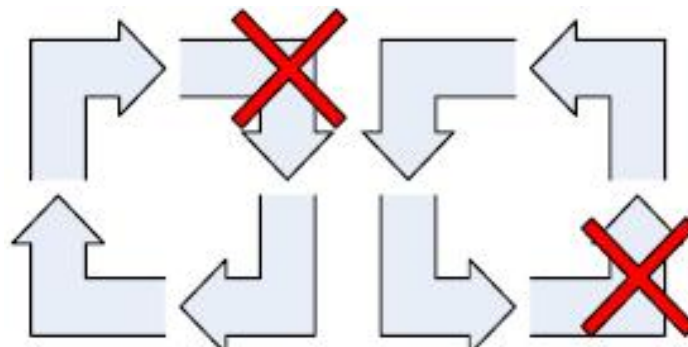


Figura 21 Giros no permitidos algoritmo odd-even columnas pares [30]

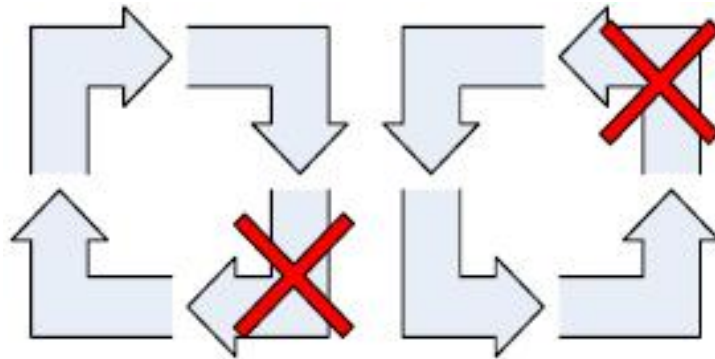


Figura 22 Giros no permitidos algoritmo odd-even columnas impares [30]

2.3.6. PARÁMETROS DE DESEMPEÑO DE UNA RED

Los principales parámetros para medir el desempeño de una red son la latencia (retraso) y el throughput.

2.3.6.1. LATENCIA

Hay que tener en cuenta, en el caso de la latencia entre más baja es mejor, Según el RFC 1242 [32] la latencia es el intervalo de tiempo que comienza cuando el final del primer bit de la trama entrante alcanza el puerto de entrada y termina cuando el comienzo del primer bit de la misma trama es visto en el puerto de salida.

Un concepto más claro, es el tiempo que una trama o paquete de datos tarda en realizar el recorrido desde la estación o nodo origen hasta su destino.

Los principales factores que influyen en la latencia de red son:

- El tamaño de la red.
- Numero de dispositivos conectados a la red.
- El tamaño de los buffers que disponen los dispositivos de conectividad.

En ocasiones es necesario conocer el tiempo que tarda la trama o paquete en ir y volver, al mismo se lo denomina Round Trip Time (RTT), es el tiempo de retardo de ida y vuelta requerido para que una señal de pulso o paquete pueda viajar de una fuente específica a un destino específico y viceversa [33].

2.3.6.2. THROUGHPUT

Según el RFC 1242 [32], el throughput es la tasa máxima a la cual ninguna de las tramas enviadas es descartada por el dispositivo.

Un concepto más explícito indica que el throughput es la tasa promedio de éxito en la entrega de un mensaje sobre un canal de comunicación. Este dato puede ser entregado sobre un enlace físico o lógico, o a través de un cierto nodo de la red, es medido en Bps (bits por segundo) [33] [34] [35] [36].

2.3.6.3. JITTER

El jitter es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Como la información se discretiza en paquetes cada uno de ellos puede seguir una ruta distinta para llegar al destino.

El jitter se define técnicamente como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. Las comunicaciones en tiempo real (como VoIP) son especialmente sensibles a este efecto. En general, es un problema frecuente en enlaces lentos o congestionados. Se espera que el aumento de mecanismos de QoS (calidad del servicio) como prioridad en las colas, reserva de ancho de banda o enlaces de mayor velocidad (100Mb Ethernet, E3/T3, SDH) puedan reducir los problemas del jitter en el futuro.

También conocido como variación de retardo, ocasiona que los paquetes de voz sean descartados por el receptor cuando este no recibe los paquetes a tiempo. [37], [38].

2.3.7. REDES ON CHIP DISPONIBLES EN EL MERCADO

Existe gran variedad de redes on chip disponibles en el mercado, lo cual indica que se tiene una gran flexibilidad al momento de elegir una, a continuación se detallan algunas de ellas, mencionando las características más relevantes. [39]

ÆTHEREAL

Fue propuesta por PHILIPS en el 2005, tiene como objetivo principal componer y pre visualizar el diseño del sistema, de esta manera se eliminan dudas que se puedan presentar en las interconexiones, proporcionando un throughput garantizado y servicios de latencia.

Características principales:

- Proporciona una reconfiguración en tiempo de ejecución.
- Consta de dos componentes: la interfaz de red y el router con múltiples conexiones entre ellos.
- La interfaz de red de esta NOC tiene cuatro puertos IP con un área de sintetizado de 0.172 mm^2 .
- Tanto el router como la interfaz de red de esta NOC están implementados con una tecnología de 0.13 um y funcionan a una frecuencia de 500Mhz .
- La interfaz de red posee un ancho de banda de 16Mb/s hacia todos los routers.
- La interfaz de red ofrece las arquitecturas de comunicación OCP y AXI.

HERMES

Desarrollada en La Pontificia Universidad Católica de Rio Grande do Sul en el año 2004. La NOC HERMES es una red directa con una topología de malla 2-D, emplea una conmutación WH, adicional usa un mínimo algoritmo de enrutamiento XY, el tamaño del flit es de 8 bits, los dos primeros flits de paquetes contienen la cabecera [21], [39].

MANGO

Mango (Message-passing Asynchronous Network-on-chip providing GS over open core protocol (OCP) interfaces), fue desarrollada en la Universidad Técnica de Dinamarca [21], [39].

- Promueve el best effort (BE) and guaranteed service (GS), el enrutamiento fuente son conexiones best effort.
- La network interface convierte OCP de dominio síncrono a OCP de dominio asíncrono.
- Los routers asignan buffers físicos por separado para los canales virtuales.

STNOC

Propuesta por ST Microelectronics en el 2002, presenta una topología anillo spidergon con trayectoria mínima, posee enlaces de 32 bit, la topología de esta NOC hace que sea muy eficiente desde el punto de vista del rendimiento.

Un ejemplo claro se muestra en la Figura 23 [39].

CHAIN

Desarrollado en la Universidad de Manchester, son ampliamente utilizados en las técnicas de circuitos asíncronos que explotan las capacidades de baja potencia [39].

Enfocado para sistemas heterogéneos de baja potencia, en el cual la red es un sistema específico.

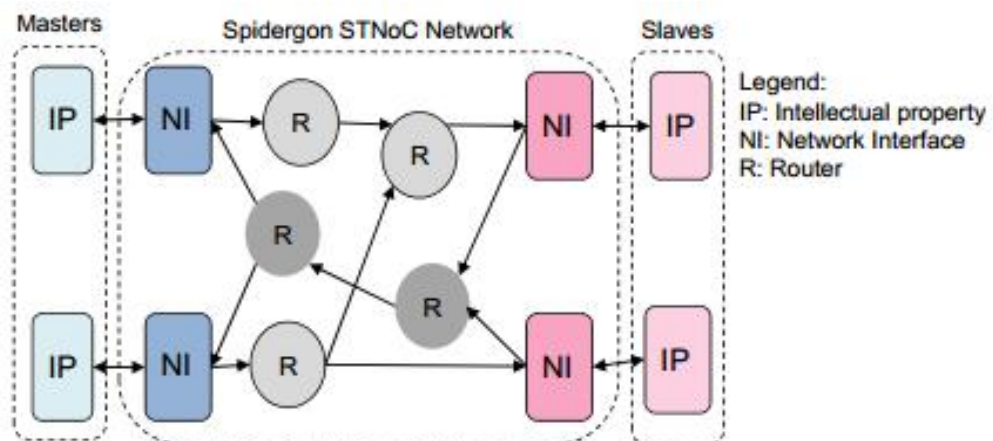


Figura 23 Ejemplo ST NOC [40]

XPIPES

Fue propuesta en una colaboración entre la Universidad de Bolonia y de la Universidad de Stanford. [21] [41]

Presenta una topología arbitraria, adaptada a la aplicación para mejorar el rendimiento y reducir costos.

2.4. NETWORK INTERFACE (NI)

La Network Interface es el componente de la NOC, encargada de realizar la conversión de la comunicación basada en paquetes de la NOC para el protocolo de alto nivel que los IP Cores utilizan, es decir permite la comunicación lógica entre los IP Cores y la NOC.

La estructura general de la Network interface se muestra en la Figura 24, donde se visualiza que la misma consta de un proceso de empaquetamiento, seguido de un transmisor asíncrono FIFO, así como de un receptor FIFO seguido de un proceso de desempaquetamiento.

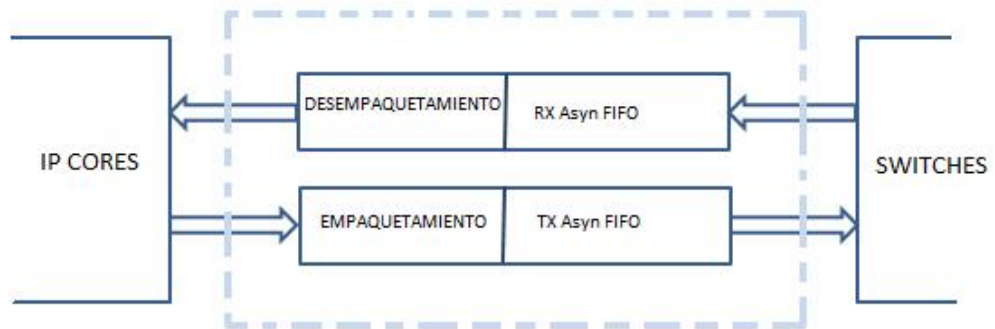


Figura 24 Estructura General Network Interface

La Network Interface cumple las siguientes funciones principales:

- Escritura/lectura de los flits a/desde el núcleo de procesamiento y el router y viceversa.
- Empaquetar las señales entrantes provenientes de los IP Cores por un número de flits ensamblados.
- Desempaquetar las señales provenientes de los switches a la especificación que tenga el IP Core.
- Transferencia de datos desde un dominio de reloj hacia otro sin tener pérdidas.
- Hacer que no se reduzca el ancho de banda en la red.
- Hacer que no se incremente la latencia impuesta en la red [42], [43], [44].

2.4.1. ARQUITECTURAS DE COMUNICACIÓN ON-CHIP

Existen dos categorías que hacen referencia a las arquitecturas de comunicación on-chip, las mismas que se describen a continuación.

2.4.1.1. ARQUITECTURA DE COMUNICACIÓN BASADA EN BUSES

La arquitectura basada en buses tiene como características principales:

- Definir la interfaz entre el IP Core y la arquitectura bus.
- En algunos casos permite definir la arquitectura bus que utiliza el protocolo de transferencia.

Algunos ejemplos de arquitecturas de comunicación basada en buses se describen a continuación, resaltando las capacidades y características de cada uno de ellas.

2.4.1.1.1. ARQUITECTURA AMBA

AMBA (Advanced Microcontroller Bus Architecture), es una arquitectura que define un estándar de comunicación para el diseño de micro controladores embebidos de alto rendimiento.

La arquitectura AMBA está compuesta de cuatro tipos de buses:

- The Advanced High-performance Bus (**AHB**)
- The Advanced System Bus (**ASB**)
- The Advanced Peripheral Bus (**APB**)

Las características principales de estos buses se describen en la Figura 25.

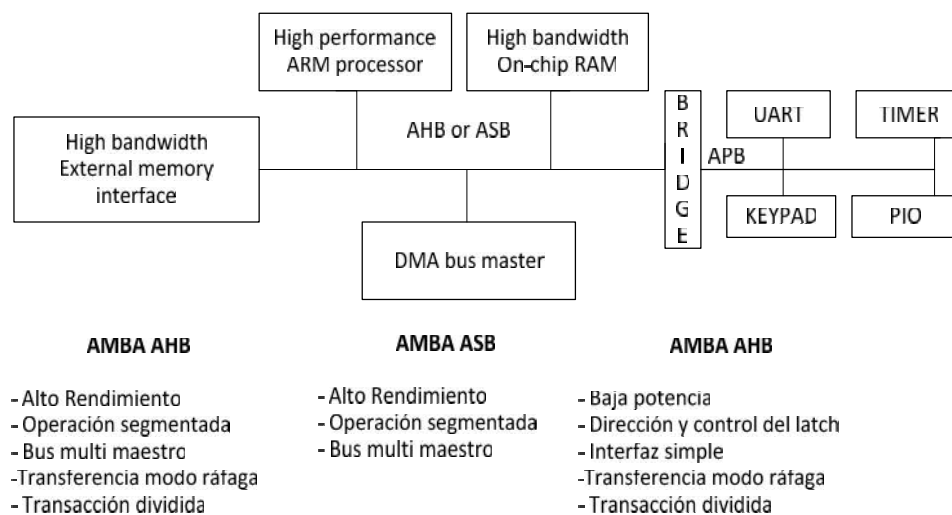


Figura 25 Arquitectura AMBA (Pasricha & Dutt, 2008)

2.4.1.1.2. ARQUITECTURA AXI

La arquitectura AXI (Advanced Extensible Interface) es una mejora de la arquitectura AMBA, también es llamada AMBA 3.0.

AXI describe una arquitectura basada en canales de alto nivel que permite una comunicación entre maestros y esclavos en un bus.

El ancho de banda del canal de datos de esta arquitectura es de 8 a 1024 bits.

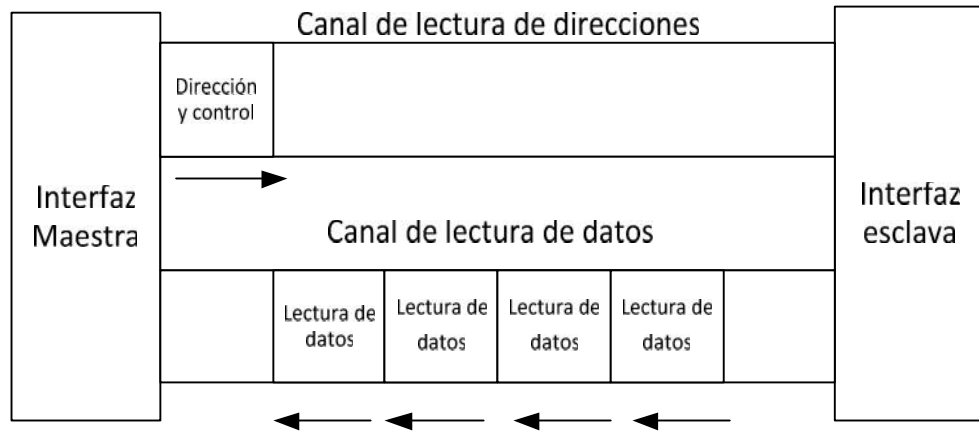


Figura 26 Canales read address y read data de la Arquitectura AXI [21]

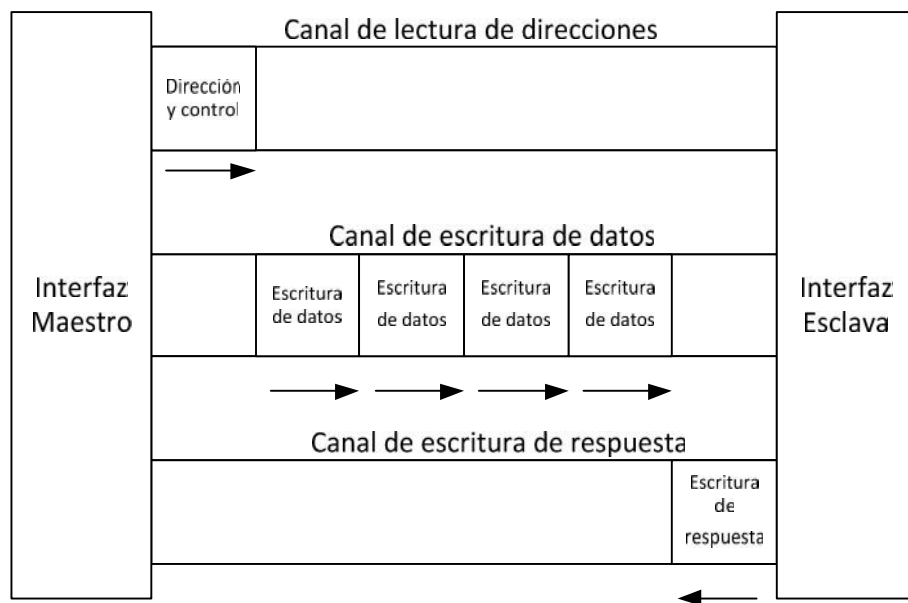


Figura 27 Canales write address, write data y write response de la Arquitectura AXI [21]

Para tener una idea más clara sobre las características más relevantes entre las arquitecturas AMBA y AXI se realiza una tabla comparativa entre ellas.

Tabla 3 Arquitectura AXI y AMBA

AMBA	AXI
Especificación basada en buses. Un solo bus de direcciones compartido, buses de lectura y escritura separados.	Especificación basada en canales, contiene cinco canales separados para leer direcciones, leer datos, escribir direcciones, escribir datos, así como una canal de respuesta de escritura que permite al esclavo responder a transacciones de escritura.
Requiere transmitir la dirección de todos elementos del dato transmitido en el bus.	Solo se necesita transmitir la dirección del primer elemento del datos transmitido en el bus, esto se logra ya que cuenta con el modo ráfaga
Esquema de transacción SPLIT sencillo, lo cual provoca que la realización de las transacciones pendiente sea muy elemental y limitada	Realiza transacción OO (out of order), ofreciendo un soporte nativo para múltiples transacciones pendientes.
No posee modo ráfaga fijo	Modo ráfaga fijo para la memoria asignada a los periféricos de salida
Apoya al soporte del cache con una seguridad básica.	Apoya al soporte del cache y una seguridad avanzada

2.4.1.1.3. OPENCORES WISHBONE

Es una arquitectura de código abierto que propone la especificación de un bus único de alta velocidad, que permita conectar todos los componentes de un diseño SOC.

Las principales características de la arquitectura WISHBONE de definen a continuación.

- Soporta el modo de operación Multimaster
- Ancho del bus de direcciones hasta 64 bits.
- Ancho del bus de datos configurable desde 8 a 64 bits (expandible).
- Soporta operaciones de lectura/escritura individuales o en bloque.
- Soporta RMW (read modify write), para realizar operaciones de tipo semáforo.
- Soporta topologías: punto a punto, flujo de datos, bus compartido o barra cruzada,
- Etiquetas definidas por el usuario, para la señalización de error sugerencias de caché, etc.
- Esquema de arbitraje flexible (prioridad fija, RR, etc).

Es una arquitectura ampliamente configurable, es decir permite al usuario personalizar etiquetas o señales que ayuden a los requisitos que demanden ciertas aplicaciones específicas.

La arquitectura de comunicación on-chip WISHBONE puede ser estructurada en diferentes topologías punto a punto, tal como se muestra en la siguiente figura. [21]

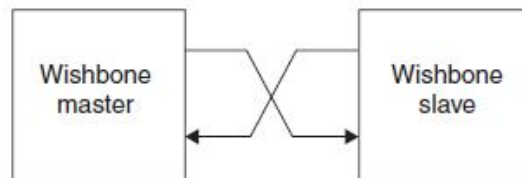


Figura 28 Topologías punto a punto de la arquitectura Wishbone [21]

2.4.2. BUS DE INTERFAZ ESTANDAR BASADO EN SOCKET

Este bus define la interfaz de un componente que se conecta a una arquitectura de comunicación basada en buses, es decir solo define la interfaz, más no la implementación de la arquitectura del bus.

Esta característica mejora la flexibilidad de reutilización del IPCore, ya que pueden ser conectados a una amplia gama de arquitecturas de comunicación basadas en buses, para lo cual estos estándares deben ser genéricos, integrales y configurables para poder tener funciones básicas y avanzadas apoyándose con la amplia gama de arquitecturas de comunicación estándar basadas en buses

En la siguiente imagen se visualiza un ejemplo de un sistema implementado con este tipo de estándar.

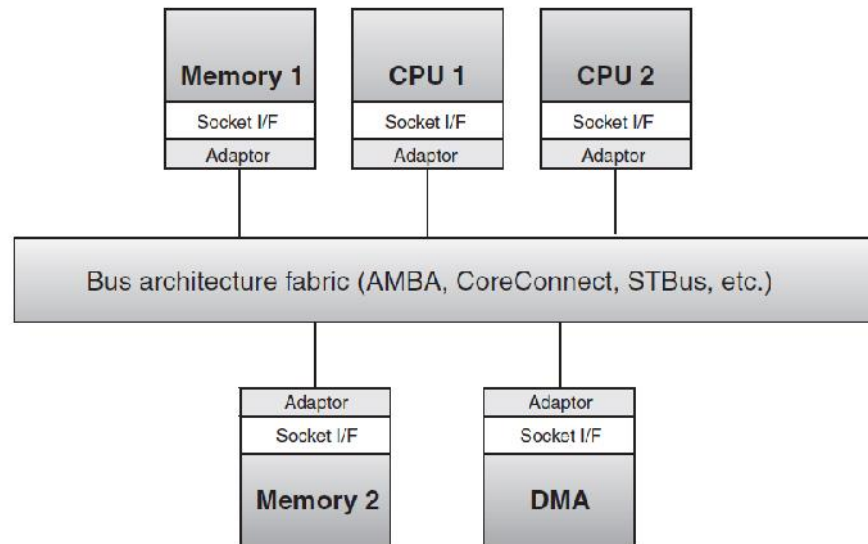


Figura 29 Ejemplo de un sistema implementado con un bus de interfaz estándar basada en SOcket [21]

En la figura anterior muestra como los diseñadores son libres de elegir cualquier estándar o arquitectura de bus como por ejemplo: AMBA, Core Connect, STBus, entre otros.

Para implementar la actual comunicación primitiva, usando interfaz estándar basada en Socket (I/F), el único requerimiento para que exista la integración entre la arquitectura del bus y el componente es el adaptador lógico de componentes que mapea la interfaz del componente a la estructura de la arquitectura del bus, a continuación se detalla una arquitectura de este tipo.

2.4.2.1.1. OPEN CORE PROTOCOL (OCP)

OCP fue lanzado en septiembre de 2003, por el grupo de trabajo OCP-IP, cuyo objetivo principal es proliferar un estándar común para las interfaces de los IP Cores, o Socket, lo cual ayuda a facilitar el “plug and play” para el diseño de los SOC, define una interfaz eficiente con un bus configurable, independiente y altamente escalable para comunicaciones en sub sistemas on chip y promueve la reutilización de los IP Cores, en los diseños SOC reduce el tiempo del mismo, así como los costos de riesgo y fabricación [16] [45], [46], [47].

La última versión de OCP es la 3.0, cuyas mejoras con respecto a las anteriores versiones son:

- **Extensiones Coherentes:** son un conjunto de comandos y señales configurables que permiten al diseñador de SOC construir una amplia gama de arquitecturas coherentes en caché (coherence cache).
- **Soporte a las nuevas señales de banda lateral:** Las señales de banda lateral se utilizan para transmitir información de control tales como: Interrupciones, reajustes, errores, e información específica de otro componente, también se utilizan para intercambiar información de estado y de control entre un componente y el resto del sistema. Por último, las señales de prueba OCP son también un conjunto de señales opcionales, y son responsables de la exploración de apoyo, control del reloj, e IEEE 1149.1 (JTAG), para fines de prueba.
- **Perfil avanzado de alta velocidad:** Este perfil está destinado para sistemas que procesan grandes cantidades en tiempo real por ejemplo los televisores digitales de alta definición.

2.5. QoS (Quality of Service)

En la actualidad el tráfico en las redes ha ido incrementándose, debido a que existen aplicaciones en tiempo que real que necesitan un ancho de banda mayor, lo cual provoca que las mismas se tornen lentas.

Para poder evitar este inconveniente y dar un buen servicio en la red es necesario configurar Calidad de Servicio dentro de la misma, con lo cual se logra dar ciertas prioridades a determinados tipos de tráfico, esta configuración es necesaria cuando la red brinda servicios en tiempo real, con el fin de optimizar el uso del ancho de banda en la red.

La calidad de servicio es un conjunto de requisitos de servicio que la red debe cumplir para asegurar un nivel de servicio adecuado para la transmisión de los datos. Estos requisitos se basan en estándares de funcionalidad de QoS [48].



Figura 30 Calidad de Servicio QoS [49]

La calidad de servicio maneja ciertos parámetros como el ancho de banda, el retardo, el jitter, la tasa de pérdidas.

En la siguiente tabla se describen los parámetros mencionados.

Tabla 4 Parámetros QoS

PARÁMETRO	UNIDADES	SIGNIFICADO
Ancho de Banda (bandwidth)	Kb/s	Indica el caudal máximo que se puede transmitir
Retardo (delay) o latencia (latency)	ms	Indica el caudal máximo que se puede transmitir
Jitter	ms	La fluctuación que se puede producir en el retardo
Tasa de pérdidas (loss rate)	%	Proporción de paquetes perdidos respecto de los enviados

Las ventajas que se tiene cuando una red es compatible con QoS puede resumirse la siguiente manera:

- La capacidad de priorizar el tráfico y, por lo tanto, permitir que los flujos importantes sean utilizados antes que los flujos de menor prioridad.

- Mayor fiabilidad en la red, debido al control de la cantidad de ancho de banda que puede utilizar una aplicación y, en consecuencia, el control sobre las carreras del ancho de banda entre aplicaciones [48], [50].

2.5.1. NIVELES DE QoS

Calidad de servicio posee tres niveles: mejor esfuerzo, servicio diferenciado y servicio garantizado, los mismos que hacen referencia a las capacidades reales de calidad de servicio de extremo a extremo

2.5.1.1. Mejor esfuerzo o best effort (BE)

Conocido como “la falta de calidad de servicio”, es decir realiza el mejor esfuerzo para entregar el paquete al destino, es una conectividad sin garantías. También se caracteriza por las de tipo FIFO, que no tienen diferenciación entre flujos. Es utilizado por aplicaciones FTP y HTTP.

2.5.1.2. Servicio diferenciado (soft QoS)

Este nivel incluye un conjunto de herramientas de clasificación y de mecanismos de cola que proveen a ciertas aplicaciones o protocolos con determinadas prioridades sobre el resto del tráfico en la red.

Se usan herramientas tales como PQ, CQ, y WFQ

2.5.1.3. PQ Cola de Prioridad

Permite que el tráfico importante consiga el manejo más rápido en cada punto donde se utiliza, es decir permite dar prioridad estricta al tráfico importante.

La cola de prioridad puede priorizar de manera flexible dependiendo del protocolo de red.

Los paquetes que llegan a la interfaz se separan en cuatro colas: baja, normal, media y alta prioridad. La salida de estas cuatro colas alimenta un búfer de transmisión de la interfaz. Los paquetes siempre van desde las colas de alta prioridad, este mecanismo se ajusta a condiciones donde existe un tráfico importante, pero puede ocasionar la falta de atención de colas de menor prioridad.

2.5.1.4. CQ Cola Personalizada

Permite al administrador priorizar el tráfico sin los efectos de la falta de atención de las colas de baja prioridad, especificando el número de paquetes o bytes que deben ser atendidos para cada cola.

Se pueden crear hasta 16 colas para categorizar el tráfico, CQ ofrece un mecanismo más refinado de encolamiento, pero no asegura una prioridad absoluta como PQ.

CQ es utilizada para proveer a tráficos particulares de un ancho de banda garantizado en un punto de posible congestión, asegurando una porción fija del ancho de banda y permitiendo al resto del tráfico utilizar los recursos disponibles.

2.5.1.5. WFQ Weighted fair queuing

Esta herramienta asigna una ponderación a cada flujo de manera que determina el orden el tránsito en la cola de paquetes. La ponderación se realiza mediante discriminadores disponibles en TCP/IP.

WFQ crea una cola separada para cada tipo de tráfico y utiliza un valor determinado para la profundidad de la cola. Está diseñado para reducir al mínimo esfuerzo de configuración, y se adapta automáticamente a los cambios de tráfico en la red.

Las herramientas PQ, CQ y WDQ sirven para administrar la congestión de la red.

2.5.1.6. Servicio Garantizado (GS):

Nivel de calidad de servicio que provee a las aplicaciones de un nivel garantizado de servicio, negociando parámetros de red de extremo a extremo. La aplicación solicita el nivel de servicio necesario con el fin de operar apropiadamente y se basa en la QoS para que se reserven los recursos de red necesarios antes de que la aplicación comience a operar [49] [41] [48] [50].

En este capítulo se realizó un estudio de arte sobre los SOC (sistemas on-chip), enfocándose en los módulos IP Cores que sirven para el diseño de los mismos. En la actualidad es necesario que las aplicaciones se ejecuten en tiempo real, tengan bajo consumo y sean multitarea entre otros, para lo cual

surgió el MPSOC (Multiprocessor System-on-Chip), se realizó una breve reseña histórica del mismo así como los diferentes tipos que existen.

En la actualidad los núcleos del MPSOC necesitan comunicarse de manera eficiente para lo cual se realizó un estudio sobre la NOC red on-chip) detallando su arquitectura, topología, conmutación, enrutamiento y los principales parámetros que miden su desempeño.

Para entender cómo se comunican los IPCores con la NOC se explicó detalles de la función de la interfaz de red que es uno de los componentes de la NOC.

En la actualidad las aplicaciones en tiempo real son las más utilizadas para lo cual se explicó el tema de Calidad de Servicio.

3. PRINCIPALES PARÁMETROS DE LA RED ON-CHIP PROPUESTA

En el presente capítulo se describe de manera detallada los principales parámetros que se deben considerar al momento de diseñar una red on-chip tales como: topología, protocolo de conmutación, algoritmo de enrutamiento, tamaño óptimo del flit, frecuencia de reloj a la cual trabaja así como el tráfico que cursa por la red, en la actualidad el tráfico en tiempo real de contenidos multimedia es el más utilizado.

3.1. ALTO RENDIMIENTO Y BAJA LATENCIA

El rendimiento es un parámetro que contiene toda red on chip y depende netamente del enrutamiento y conmutación que tenga la misma, así como la topología que se esté usando.

El enrutamiento hace referencia a que la carga se encuentre perfectamente equilibrada sobre los diferentes caminos alternativos que tenga la red.

Para aquello se debe escoger un algoritmo de enrutamiento que mejor se acople a las necesidades de la red que se está usando.

El algoritmo de conmutación y enrutamiento que se utilice es primordial para obtener un alto rendimiento y una baja latencia.

Recordar que el algoritmo de conmutación determina cuando y como se establecen las conexiones internas entre entradas y salidas, así como que mensajes deben utilizar las rutas establecidas.

Para elegir el protocolo de comunicación hay que tener en cuenta los siguientes aspectos: [26], [51].

- Estar enfocado a la reutilización de los IP Cores,
- Facilidad de la integración de IP Cores
- Escalabilidad y flexibilidad

Algunos protocolos de comunicación descritos en capítulos anteriores son:

- **OCP** Open Core Protocol
- **AXI** Advanced Extensible Interface
- **AMBA** ARM microcontroller bus architecture
- **WISHBONE**

3.2. PARÁMETROS DE DISEÑO DE UNA RED ON CHIP

Dentro de una red on chip, es necesario tener los valores óptimos de ciertos parámetros que describen el desempeño que posee la misma, es decir la red on chip debe ofrecer garantías sobre la latencia y throughput, considerando que la misma debe soportar diferentes tipos de tráfico, enfocándose en la función que cumple la interfaz de red dentro de la red on chip, para lograr aquello se debe tomar en cuenta los principales parámetros de diseño de una red on chip.

Enfocándose en la interfaz de red, la misma debe brindar servicios a la capa de transporte del modelo de referencia OSI, ya que es la primera capa en donde los servicios ofrecidos son independientes de la implementación de la red, esto es esencial para lograr el desacoplamiento entre el cómputo y la comunicación, lo cual permite que los módulos IP y sus interconexiones ser diseñados independientemente el uno del otro. Existen varias arquitecturas que se pueden usar para cumplir con este fin, tales como: como VCI (interfaz de componente virtual), OCP (Open Core Protocol) y AXI (Advanced Extensible Interface).

El presente estudio propone una interfaz de red compatible con las arquitecturas de comunicación más comunes, es decir su controlador tenga una configuración flexible, el tiempo de diseño sea corto y ofrezca un flujo de datos de alto rendimiento sin problemas, así como debe permitir conectar varios IPCores con los routers

El objetivo de la presente propuesta es identificar los temas clave de diseño y desarrollo de una interfaz de red eficiente, para que pueda ser utilizada con diferentes tipos de topologías NOC tales como: Malla 2D, TORUS, Spidergon u Octagon. [52].



Figura 31 Modelo de Referencia OSI [53]

A continuación se presentan los principales parámetros de diseño para una red on chip de acuerdo a las prestaciones que se desee.

3.2.1. TOPOLOGÍA DE LA RED

Como se describió la topología es la descripción de las interconexiones en la red. Entre las opciones a elegir en forma general tenemos: Topología de redes directas, topología de redes indirectas, tipología de redes irregulares, cada una de estas topologías tiene sus diferentes variantes.

Una topología tiene parámetros tales como el grado, el diámetro, la complejidad del enlace, ancho de bisección, etc.

A continuación se describen con mayor detalle estos parámetros.

3.2.1.1. Grado del Nodo

Es el número de enlaces conectados a un nodo, si todos los nodos de la red tienen el mismo grado se trata de una red regular, caso contrario es una red irregular, el grado de un nodo describe la complejidad de E/S de un nodo, el mismo puede ser constante o variar con el tamaño de una red.

La característica más recomendable consiste en tener una topología de cuyo grado sea fijo y menor. Al tener un menor grado ayuda al rendimiento en los enlaces, así como un grado fijo ayuda a reducir el esfuerzo para añadir nuevos nodos a la red existente, es común que en las topologías se limite el grado del nodo.

Consideraciones como la escalabilidad de la red, así como la complejidad en el espacio limitan el número de nodos con los que cada nodo puede comunicarse directamente.

3.2.1.2. Diámetro

Es la máxima distancia entre dos nodos en una red., si no existe conexión directa entre dos nodos, el mensaje tiene que viajar a través de algunos nodos intermedios, lo cual introduce retrasos de salto múltiple, ya que el retraso es proporcional a la cantidad de saltos.

3.2.1.3. Complejidad del enlace

Determinada por el número total de enlaces en la topología.

Al agregar más enlaces a una red, puede reducir su diámetro y brindar una mejor comunicación entre los nodos.

Hay que tomar en cuenta que los enlaces generan costos, ya que una mayor complejidad de los enlaces induce mayor complejidad del hardware y área.

3.2.1.4. Ancho de Bisección

Es el número de enlaces necesarios eliminar para dividir la topología en dos redes con un tamaño aproximadamente igual, se desea un ancho de bisección grande, ya que brinda más rutas entre dos subredes y por lo tanto mejora el rendimiento general.

La topología usada fue la tipo malla en donde todos los enlaces tienen la misma longitud, la conexión al nodo y el algoritmo de enrutamiento es simple, lo cual facilita la implementación física.

3.2.2. PROTOCOLO DE CONMUTACIÓN

Se debe elegir el protocolo de conmutación a usar, en capítulos anteriores se describió los diferentes protocolos existentes, con esta información y enfocándose a los requerimientos de la network interface deseada, se decidió

usar el algoritmo *Wormhole*, ya que el mismo realiza la transmisión del paquete mediante flits, lo cual ayuda considerablemente a reducir la latencia en la red, el funcionamiento se detalla en capítulos anteriores.

La ventaja principal de este algoritmo es que usa de forma eficiente los canales físicos, ya que posee canales virtuales que evitan que se produzcan situaciones de bloqueo, dichos bloqueos ocurren cuando el tamaño del paquete enviado es demasiado grande, por cuál debe ser repartido a dos o más routers, es decir el paquete permanece en la red ocupando recursos, si otro mensaje desea ocupar el mismo recurso también se bloqueará. [54].

3.2.3. ALGORITMO DE ENRUTAMIENTO

Con respecto a elegir el algoritmo de enrutamiento que se usó, se enfocó haciendo referencia a obtener valores de latencias, rendimiento y utilización del canal óptimos, el algoritmo de enrutamiento *odd-even* fue utilizado en el presente proyecto, ya que el mismo tiene una mejora del 18 y 25 por ciento frente a los otros algoritmos con respecto al rendimiento [55].

Este algoritmo brinda los valores más altos de rendimiento por paquete al compararlo con otros algoritmos de enrutamiento tales como: algoritmo origen y algoritmo XY. [56].

Adicional la técnica de enrutamiento que realiza *odd-even* ayuda a evitar el *dead-clock*, que es el estancamiento de paquetes debido a que los mismos esperan el uno del otro para liberar recursos.

3.2.4. TAMAÑO FLIT

El tamaño del flit influye en los parámetros de desempeño de la red, si el mismo es alto ofrece un mayor desempeño de la red, pero el costo de la misma de igual forma se incrementa, por lo cual el tamaño del flit debe ir de acuerdo a las necesidades de la NOC, sin descuidar el tema de costo.

El costo que involucra mejorar el rendimiento al aumentar el tamaño del flit no es compensado, por lo cual es difícil justificar un incremento en el tamaño del flit. Para mejorar el rendimiento y latencia de la red on chip, el valor del flit debe ser igual o menor que el tamaño de paquete más pequeño. [57].

3.2.5. FRECUENCIA DE RELOJ

La frecuencia a la cual trabaja la red on chip, incide directamente en el desempeño de la red, pero hay que considerar los efectos que implicarían trabajar a una alta frecuencia, sin duda a una alta frecuencia de trabajo aportaría un mayor rendimiento, sin embargo tanto el consumo de potencia así como la complejidad del cableado aumentarían, hay que considerar que si se desea implementar la red on chip estudiada, se debe tomar en cuenta la frecuencia máxima de operación de la FPGA a utilizar, la frecuencia de reloj a la cual trabaja la tarjeta Virtex-6 ML605 que se dispone en la Universidad de las Fuerzas es de 700 MHz [58].

3.2.6. TRÁFICO DE LA RED

3.2.6.1. CBR (Tasa de bits constante)

Este tipo de tráfico está destinado para aplicaciones en tiempo real, ya que posee restricciones de retardo y tiene un patrón diferente al tráfico *bursty* (ráfaga) que varía en el tiempo, el CBR reserva con exclusividad los recursos de la red, es decir que un ancho de banda constante es puesto a disposición durante la conexión, en caso de tener un tráfico que presente variación en su ancho de banda no es recomendable usar un tráfico CBR ya que se desperdiciaría el mismo [59].

3.2.6.2. TRÁFICO MULTIMEDIA

En la actualidad se han incrementado el uso de aplicaciones multimedia, específicamente la transmisión de video a través de una red on chip, lo cual ha dado lugar a varias implementaciones de hardware de video y códec de audio para un procesamiento más rápido, por lo cual evaluar la NOC bajo un tráfico multimedia se convierte en un aspecto importante al momento de estudiar los parámetros de desempeño de una red on chip.

Existen limitantes al momento de transmitir este tipo de tráfico, debido a la calidad de los videos, hacen que la información a transmitir por medio de la red sea en mayor cantidad de bit, si no se cuenta con una infraestructura optima la calidad de los videos se verá afectada ocasionando fallas en las imágenes, pixelaciones o congelamientos ocasionando retrasos en la transmisión [60].

Para el presente proyecto se simulará un tráfico de video MPEG-4, que es un estándar diseñado para admitir el transporte, la sincronización, la codificación y el procesamiento de contenidos multimedia transmitidos en línea, algunas áreas de aplicación son: Tv digital, contenidos multimedia para móviles, video en la red, video conferencia entre otros [61].

MPEG-4 ofrece una alta compresión de datos ocupando menor espacio de memoria sin descuidar la calidad del video [61].

Para poder analizar este tipo de tráfico en la red on chip, es necesario analizar lo referente al parámetro *Hurst*.

3.2.6.3. PARÁMETRO HURST

El parámetro *Hurst* mide el grado de auto semejanza de las señales a través de procesos estocásticos, lo cual permite representar tráfico de video para poder simular el mismo dentro de una red on chip.

El valor del parámetro *hurst* varía entre 0 y 1, en la Tabla 5 se detalla este rango.

Tabla 5 Rangos Parámetro Hurst

RANGO VALOR PARÁMETRO HURST	TIPO DE SERIE
0.8 – 1	Películas, series y dibujos animados
0.6 – 0.8	Transmisiones deportivas
> 1	Video conferencia

[62]

Hay que tomar en cuenta que cuando las escenas de un video tienen mucha variación en movimiento, brillo, color entre otros el parámetro *hurst* aumenta, un video en alta definición tiene un parámetro mayor que un video de menor calidad.

Se han tomado los valores del parámetro *hurst* ya calculados que representan el tráfico de conocidas series de televisión y películas que atravesarán la red on chip, así como el valor máximo y mínimo de las tramas que componen la traza de video en la Tabla 6 de detallan algunos ejemplos [62].

Tabla 6 Valores parámetro hurst, máxima y mínima trama de la traza de series de tv

TRAZA	PARÁMETRO HURST	LONGITUD MÍNIMA DE TRAMA DE LA TRAZA (bytes)	LONGITUD MÁXIMA DE TRAMA DE LA TRAZA (bytes)
ARD news	0.967	123	17055
Diehard III	0.904	26	8161
Formula 1	0.736	26	7032
Futurama	0.745	27	19224
From dusk till dawn	0.909	74	15745
First contact	0.828	26	5852
Mr. Bean	0.816	26	7565
Jurassic Park	0.881	26	8154
VIVA video clips	0.904	31	9025
N3 talk	0.882	106	17062
Silence of the lambs	0.935	26	11292
Simpsons	0.769	0	20064
Star wars IV	0.770	26	4686

[63]

En la siguiente imagen se detalla el proceso que se debe seguir para el diseño de un sistema on chip de manera general.

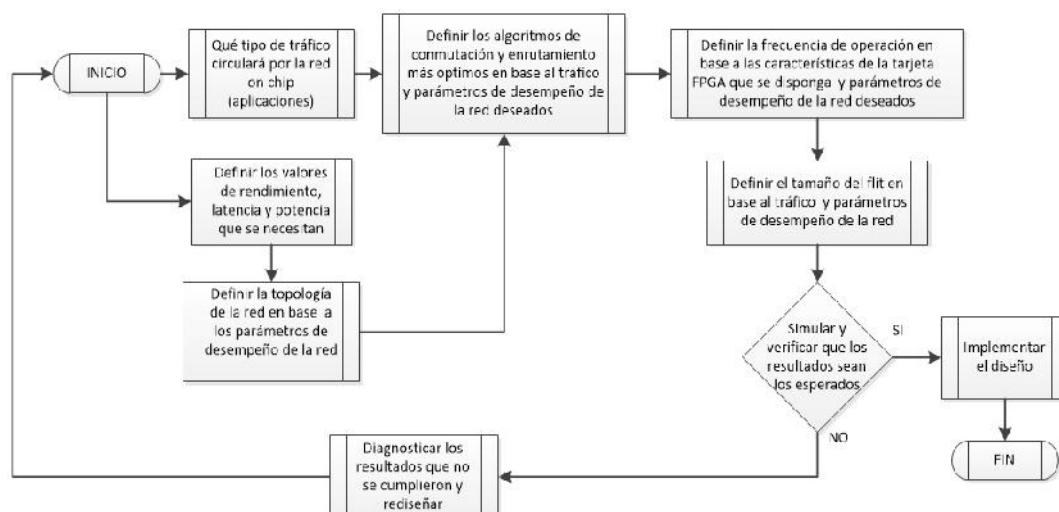


Figura 32 Proceso de un diseño de un sistema on chip

4. SIMULACIÓN

Para verificar que las configuraciones de la red on-chip tengan los resultados deseados se procedió a simular la misma, obteniendo datos de latencia, rendimiento, potencia que consume tanto para un tráfico CBR como multimedia.

Para realizar la simulación de la NOC propuesta con las configuraciones mencionadas en el capítulo anterior, en donde se pueda obtener valores aproximados de throughput, latencia y consumo de potencia que tiene la misma, se usó el simulador NIRGAM versión 2.1, debido a que el mismo reúne las especificaciones de topología, algoritmos de conmutación y enrutamiento que se requiere.

La NOC simulada fue configurada tomando en cuenta el análisis realizado en el capítulo anterior, algunos parámetros dependen del tráfico que circula por la red on chip.

4.1. SIMULACIÓN TRÁFICO CBR (Constant Bit Rate)

A continuación se detalla los parámetros con los cuales se simularon red on chip con un tráfico CBR (constant bit rate).

Tabla 7 Parámetros de configuración simulación con tráfico CBR

PARÁMETRO	ESPECIFICACIÓN
topología	mallá 5x5
algoritmo de conmutación	Wormhole
algoritmo de enrutamiento	odd-even
tamaño del flit	6/11/21 bytes
tamaño paquete	1 kbytes
uso del ancho de banda	75%
frecuencias de operación	500Mhz-700Mhz 1Ghz
ciclos de reloj que dura la simulación	1000

La topología malla 5x5 que se simuló Figura 32, consta de 25 azulejos que intercambian datos entre sí.

En la Figura 32 se visualiza los azulejos origen y destino de la red, así como la trayectoria que se tiene para llegar de uno al otro respectivamente.

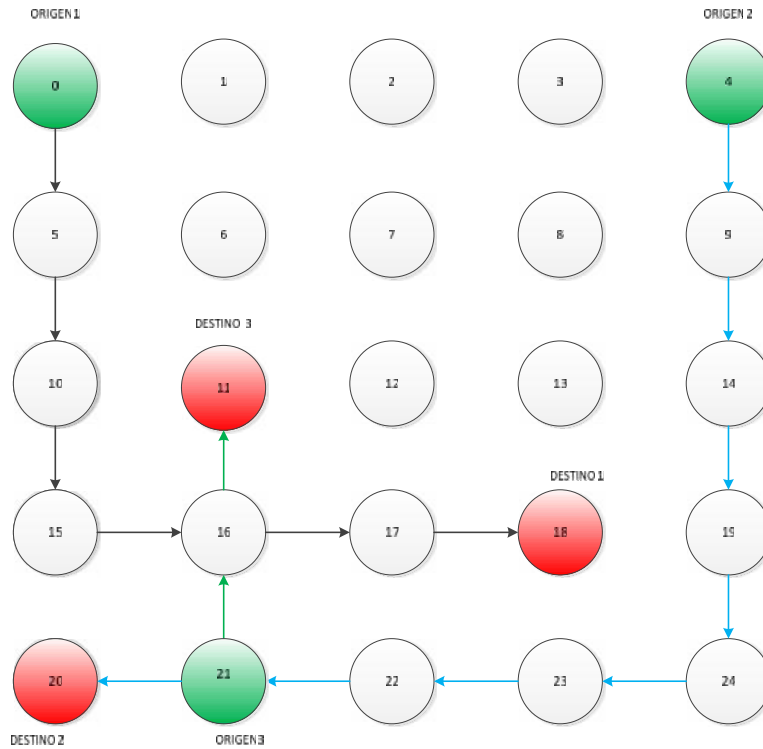


Figura 33 Red on chip simulada con tráfico CBR

En la Figura 33 se detalla los elementos que contienen cada azulejo.

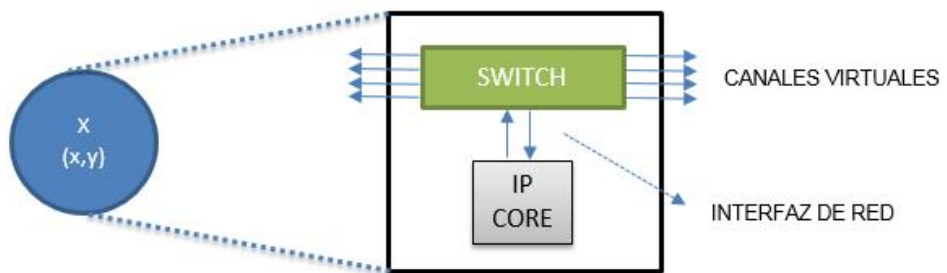


Figura 34 Elementos azulejo red on-chip [64]

En la tabla 8 se muestran los parámetros obtenidos de la simulación variando el tamaño del flit con una frecuencia de 500Mhz.

Tabla 8 Parámetros obtenidos tráfico CBR con frecuencia de operación de 500Mhz.

PARÁMETRO	VALOR(flit)		
	6 bytes	11 bytes	21 bytes
ciclo de reloj	2 ns	2 ns	2 ns
latencia promedio por flit (canal)	0.137 ns	0.274 ns	0.547 ns
latencia promedio por paquete(canal)	33 ns	32.909 ns	32.864 ns
latencia promedio por flit (total)	232.46 ns	225.194 ns	210.716 ns
rendimiento	7.438 Gbps	13.588 Gbps	25.699 Gbps
flits generados	801	798	765
flits recibidos	706	704	704
potencia consumida	0.115 watts	0.279 watts	0.796 watts

En las Tablas 9 y 10 se visualizan los resultados obtenidos con una frecuencia de 700Mhz y 1Ghz respectivamente.

Tabla 9 Parámetros obtenidos tráfico CBR con frecuencia de operación de 700Mhz.

PARÁMETRO	VALOR (flit)		
	6 bytes	11 bytes	21 bytes
ciclo de reloj	1.43 ns	1.43 ns	1.43 ns
latencia promedio por flit (canal)	0.1 ns	0.20 ns	0.40 ns
latencia promedio por paquete(canal)	24.10 ns	24,11 ns	24.11 ns
latencia promedio por flit (total)	156.47 ns	149,6 ns	135.41 ns
rendimiento	10 Gbps	18.26 Gbps	34.52 Gbps
flits generados	798	798	765
flits recibidos	708	705	705
potencia consumida	0.148 watts	0.36 watts	1.04 watts

Tabla 10 Parámetros obtenidos tráfico CBR con frecuencia de operación de 1Ghz.

PARÁMETRO	VALOR (flit)		
	6 bytes	11 bytes	21 bytes
ciclo de reloj	1 ns	1 ns	1 ns
latencia promedio por flit (canal)	0.070 ns	0.140 ns	0.28 ns
latencia promedio por paquete(canal)	16.857 ns	16.86 ns	16.857 ns
latencia promedio por flit (total)	113.353 ns	109.99 ns	103.04 ns
rendimiento	14.29 Gbps	26.09 Gbps	49.320 Gbps
flits generados	801	798	765
flits recibidos	708	705	705
potencia consumida	0.205 watts	0.506 watts	1.465 watts

4.2. SIMULACIÓN TRÁFICO MULTIMEDIA

La segunda simulación se realizó inyectando tráfico multimedia a la red, representando a varias series de televisión descritas anteriormente, para esta simulación se configuró los siguientes parámetros.

Tabla 11 Parámetros de configuración simulación con tráfico multimedia

PARÁMETRO	ESPECIFICACIÓN
topología	mallla 5x5
algoritmo de conmutación	Wormhole
algoritmo de enrutamiento	odd-even
tamaño del flit	9/16/26 bytes
frecuencias de operación	500Mhz-700Mhz 1Ghz
ciclos de reloj que dura la simulación	20000

Una vez realizada la simulación con tráfico multimedia, se obtuvieron los siguientes datos, en las tablas 12, 13 y 14 se muestran los resultados de la simulación con una frecuencia de operación de 500 MHz, 700Mhz y 1Ghz respectivamente.

Tabla 12 Parámetros obtenidos tráfico multimedia con frecuencia de operación de 500Mhz.

PARÁMETRO	VALOR(flit)		
	9 bytes	16 bytes	26 bytes
ciclo de reloj	2 ns	2 ns	2 ns
latencia promedio por flit (canal)	0.094 ns	0.16 ns	0.286 ns
latencia promedio por paquete(canal)	34.709 ns	33.608 ns	33.872 ns
latencia promedio por flit (total)	241.186 ns	240.948 ns	240.366 ns
rendimiento	9.301 Gbps	16.346 Gbps	26.45 Gbps
flits generados	15091	15119	15177
flit recibidos	14998	15025	15084
potencia consumida	0.204 watts	0.504 watts	1.149 watts

Tabla 13 Parámetros obtenidos tráfico multimedia con frecuencia de operación de **700Mhz.**

PARÁMETRO	VALOR (flit)		
	9 bytes	16 bytes	26 bytes
ciclo de reloj	1.43 ns	1.43 ns	1.43 ns
latencia promedio por flit (canal)	0.062 ns	0.11 ns	0.215 ns
latencia promedio por paquete(canal)	24.106 ns	24.21 ns	23.89 ns
latencia promedio por flit (total)	172.491 ns	172.28 ns	171.83 ns
rendimiento	12.881 Gbps	22.818 Gbps	38.694 Gbps
flits generados	15087	15117	15184
flit recibidos	14994	15024	15091
potencia consumida	0.276 watts	0.688 watts	1.688 watts

Tabla 14 Parámetros obtenidos tráfico multimedia con frecuencia de operación de **1Ghz.**

PARÁMETRO	VALOR (flit)		
	9 bytes	16 bytes	26 bytes
ciclo de reloj	1 ns	1 ns	1 ns
latencia promedio por flit (canal)	0.041 ns	0.072 ns	0.127 ns
latencia promedio por paquete(canal)	16.719 ns	17.016 ns	16.573 ns
latencia promedio por flit (total)	120.639 ns	120.493 ns	120.291 ns
rendimiento	19.367 Gbps	33.197 Gbps	53.397 Gbps
flits generados	15084	15113	15160
flit recibidos	14991	15020	15066
potencia consumida	0.385 watts	0.965 watts	2.23 watts

4.3. ANÁLISIS DE RESULTADOS

Con los datos obtenidos del simulador NIRGAM se analizan los parámetros de desempeño de red obtenidos en base a la configuración realizada sobre el simulador.

RENDIMIENTO, LATENCIA Y POTENCIA

El análisis se enfoca en los principales parámetros de desempeño de una red on chip: rendimiento, latencia y potencia.

El tamaño del flit influye en estos parámetros, si aumenta el tamaño del flit, se observa un mejor rendimiento así como un incremento en la potencia que disipa la red, por tal motivo, se debe tomar en cuenta el valor de este parámetro de tal manera que no afecte los valores óptimos de trabajo de la red on Chip. La latencia, a su vez, no sufre considerables cambios.

La frecuencia a la cual trabaja la red es de 700 MHz (frecuencia máxima que soporta la tarjeta FPGA Virtex-6 ML605), se verifica que por cada 100 MHz la potencia que consume la red se incrementa en 0.2142 watts, por tal motivo si se desea tener un alto rendimiento que implicaría altas frecuencias, la potencia es un parámetro a tener en cuenta. La latencia en este caso, tampoco muestra un comportamiento que verifique cambios.

En la Figura 34 y Figura 35 se muestra que el comportamiento del rendimiento variando el tamaño del flit tanto con tráfico CBR como con tráfico multimedia es de manera proporcional.

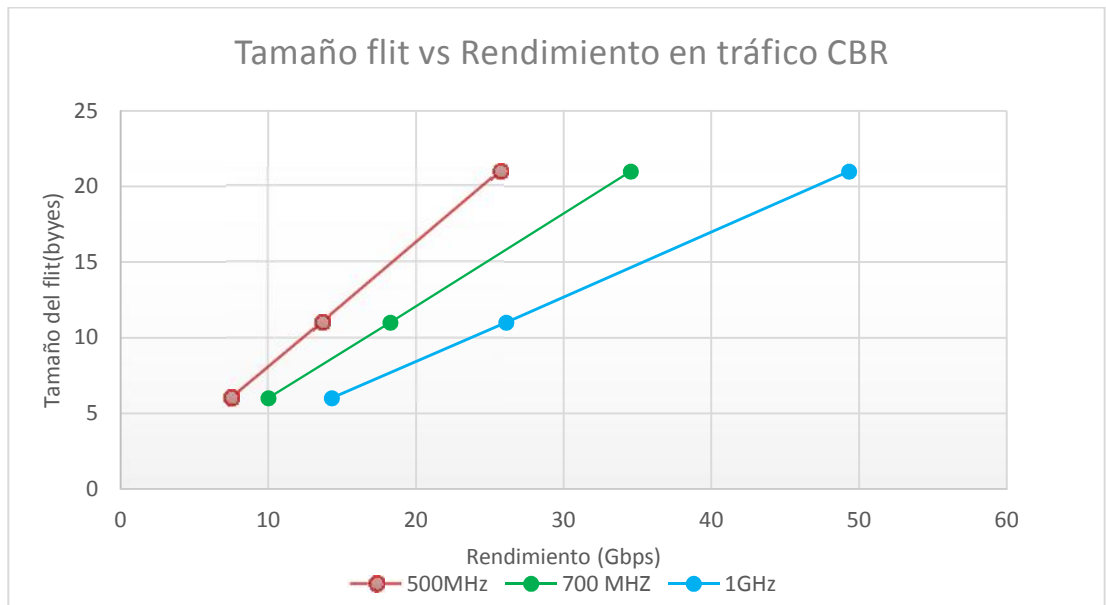


Figura 35 Tamaño flit vs Rendimiento en tráfico CBR con diferentes frecuencias de operación.

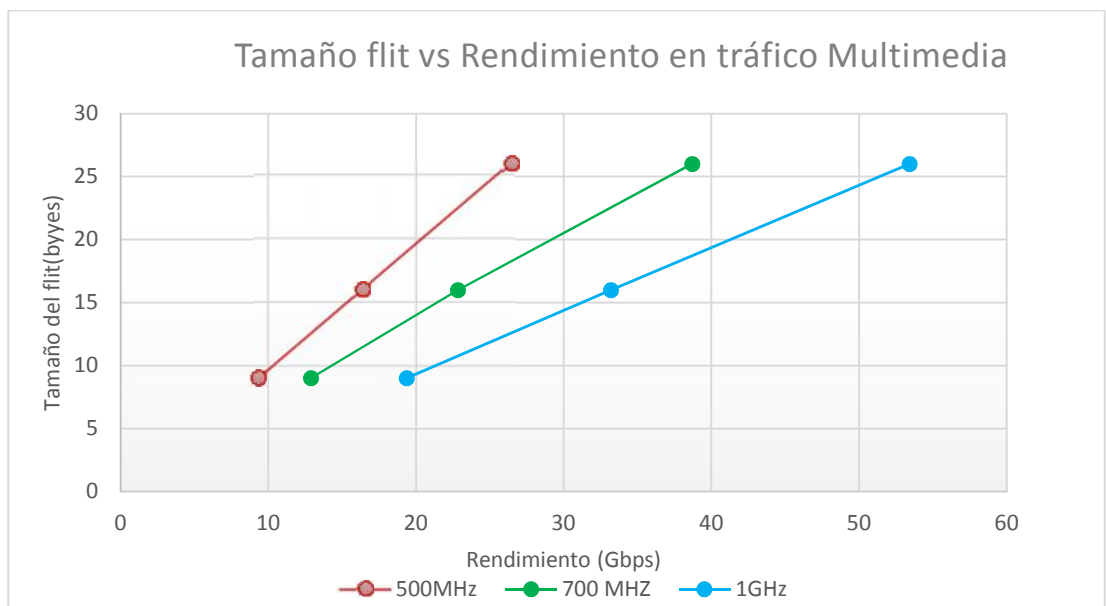


Figura 36 Tamaño flit vs Rendimiento en tráfico multimedia con diferentes frecuencias de operación.

La Figura 36 y Figura 37 muestra que al tener un mayor rendimiento el consumo de potencia se incrementa de manera proporcional, por lo cual es importante analizar estos dos parámetros a la par.

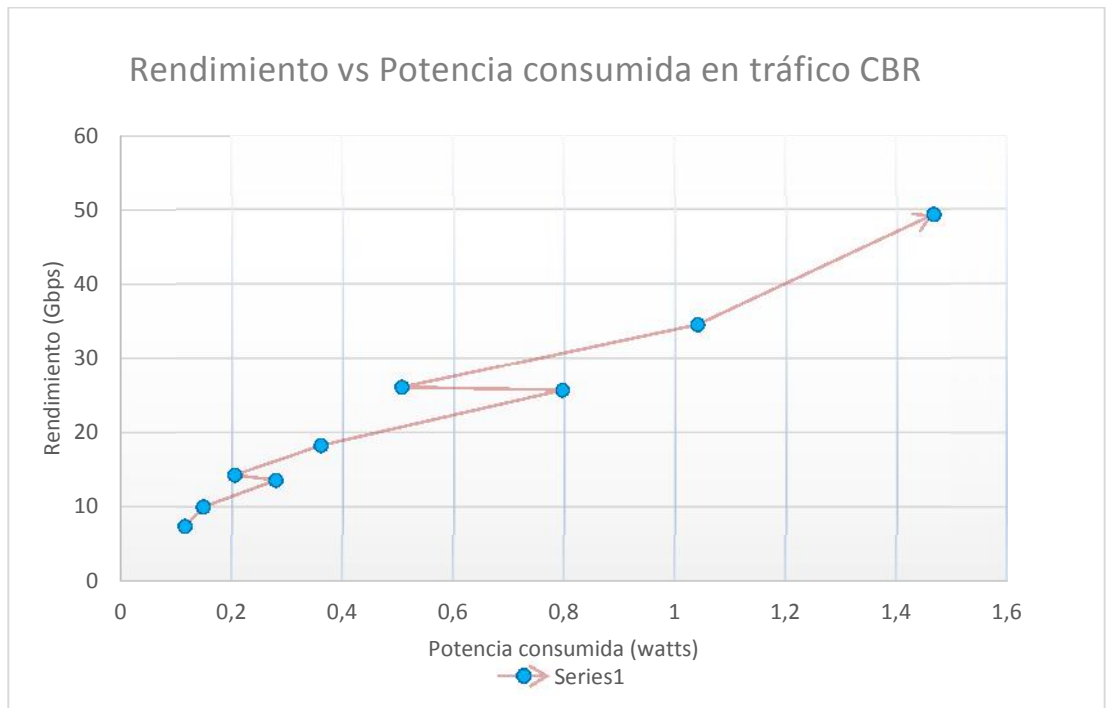


Figura 37 Rendimiento vs Potencia consumida en tráfico CBR

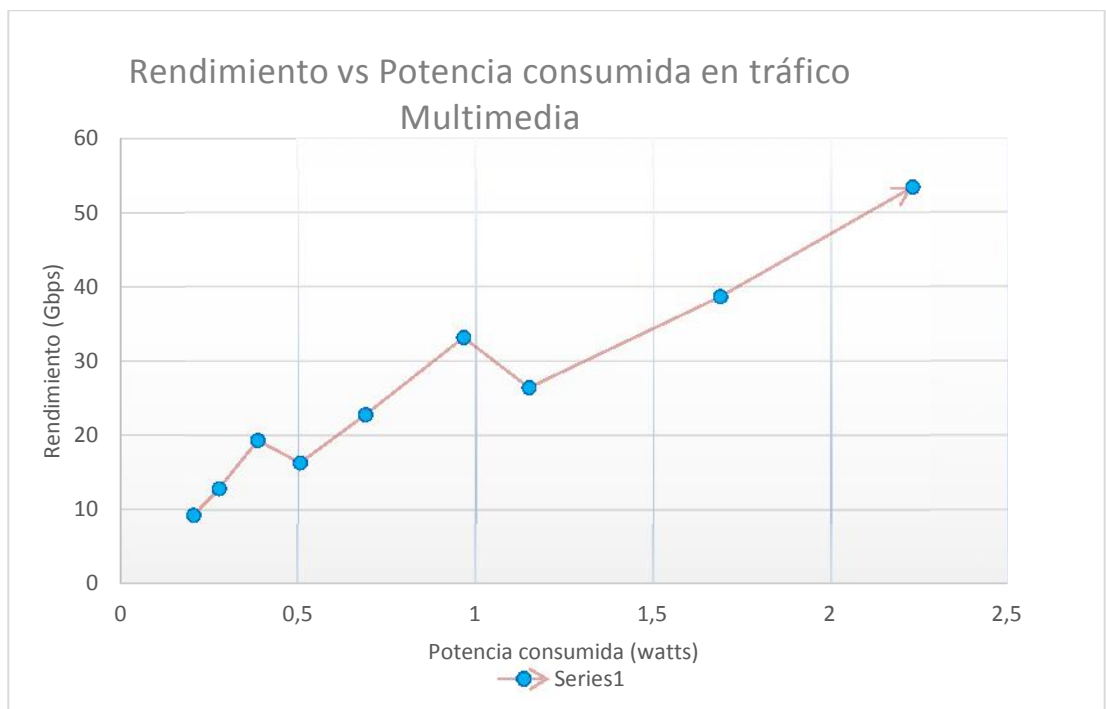


Figura 38 Rendimiento vs Potencia consumida en tráfico multimedia

En la Figura 38 y Figura 39 se verifica que la latencia varía en base al rendimiento de la red en mínima proporción, a un mayor rendimiento la latencia se reduce, por lo cual al tener una mayor frecuencia de operación la latencia de la red disminuye.

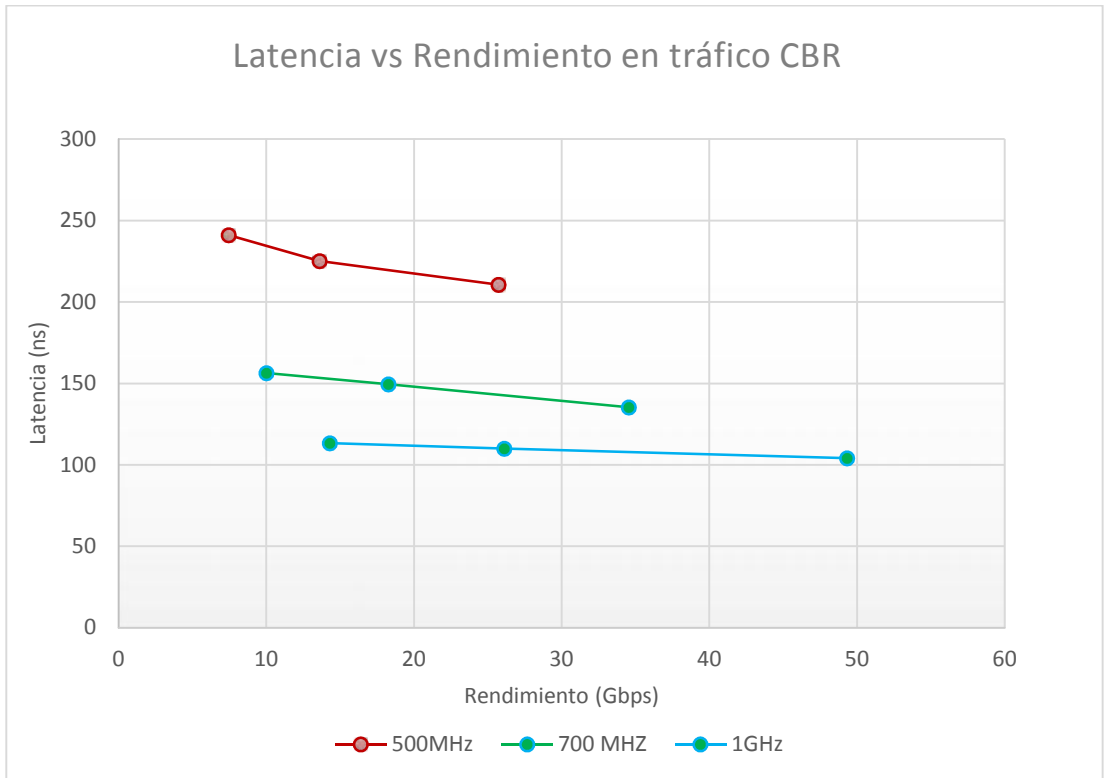


Figura 39 Latencia vs Rendimiento en tráfico CBR

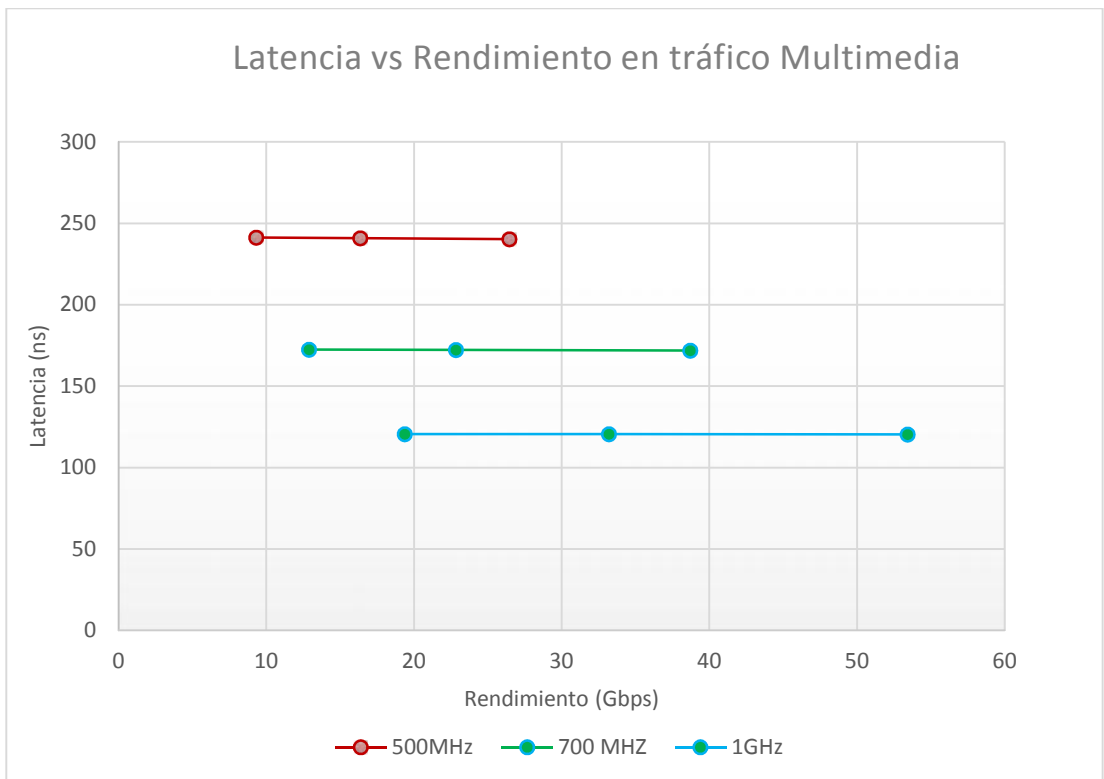


Figura 40 Latencia vs Rendimiento en tráfico multimedia

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se ha llegado a las siguientes conclusiones

Conclusiones

- Se realizó el estudio y comparación de los algoritmos de conmutación más usados en las redes on chip, el algoritmo de conmutación Wormhole fue utilizado en el presente proyecto, el mismo que reduce el tamaño de los buffers debido a que los paquetes son transmitidos en unidades llamadas flit (unidades de control de flujo) que son las unidades más pequeñas para el control de flujo. La cabecera del flit es la única que posee la ruta y los demás paquetes de flits que llevan los datos siguen el camino determinado por esta cabecera.

- Referente a los algoritmos de enrutamiento más comunes en las redes on chip, se debe elegir tomando en cuenta ciertos parámetros: la potencia que se requiere para realizar el enrutamiento, la misma que debe ser la menor posible, el rendimiento de la NOC que puede aumentarse maximizando la utilización del tráfico en la red así como reduciendo el retardo, y finalmente la robustez para adaptarse a los cambios de tráfico es un parámetro.

- Se utilizó el simulador NIRGAM para verificar los principales parámetros de desempeño de una red on chip, específicamente el rendimiento, latencia y consumo de potencia de la misma.

- Con los resultados de la simulación, se verifica que el tamaño del flit es directamente proporcional al rendimiento de la NOC, al reducir el tamaño del flit, el rendimiento se redujo de igual manera, pero hay que tomar en cuenta que si se aumenta el tamaño del flit, el costo de implementación aumenta.

- Se verifica que el rendimiento de la red está estrechamente relacionado con el tamaño del flit, la frecuencia de operación y el consumo de potencia, lo cual se debe tener en cuenta si es factible variar cualquiera de estos parámetros.

- La frecuencia máxima de operación a la cual trabaja la tarjeta FPGA Virtex-6 ML605 que se dispone en la Universidad de las Fuerzas Armadas ESPE es de 700Mhz, lo cual se consideró al momento de simular la red on-chip.

Recomendaciones

- Si se desea realizar una implementación sobre una tarjeta de desarrollo FPGA que soporte aplicaciones en tiempo real, se debe considerar las especificaciones que tenga la misma, enfocándose en la frecuencia máxima de operación, así como en el almacenamiento que brinda la misma.
- Para implementaciones de sistemas embebidos, se recomienda utilizar tarjetas de desarrollo Altera que a más de tener un costo menor a Xilinx ofrece una interfaz de programación sencilla y ofrece una línea de productos mucho más amplia.
- Se recomienda realizar la implementación de este y otros estudios realizados sobre la tarjeta de desarrollo altera de la serie STRATIX.
- Se recomienda realizar un estudio más profundo referente al parámetro hurst, con el fin de evaluar las funcionalidades del mismo en aplicaciones multimedia.

ANEXOS

GLOSARIO DE TÉRMINOS

Conmutación WH: Técnica de Conmutación en donde el requerimiento de buffer se reduce a un flit en lugar del paquete completo.

Coherence Cache: es la consistencia que se tiene a que los datos de recurso compartido se encuentren almacenados en múltiples cachés locales.

Deadlock: cuando un grupo de paquetes no pueden avanzar porque están esperando el uno del otro para liberar recursos

Flit: Es la unidad de transmisión mínima dentro de la NOC en un ciclo de reloj.

FPGA Field Programmable Gate Array: es un dispositivo semiconductor que posee bloques lógicos interconectados para que puedan ser programados

HDL Hardware Description Language: Lenguaje que describe el hardware de los sistemas digitales en forma textual.

JTAG- IEEE 1149.1: es una arquitectura eficiente que facilita la depuración de errores, así como las pruebas de funcionamiento para dispositivos digitales

Livelock: Ocurre cuando los paquetes enviados continúan moviéndose a través de la red, pero no avanzan hacia el destino, los mismos cambian de estado constantemente haciendo referencia a que desean un mismo recurso de red.

MPSOC Multiprocessor System-on-Chip: Sistema on-chip que usa múltiples procesadores utilizado generalmente en sistemas embebidos.

Netlist: Representación en lenguaje de descripción de hardware de la conectividad de un circuito

Plug and Play: estándar para añadir hardware que requiere que un dispositivo se identifique a sí mismo al conectarse, es decir ayuda a configurar los dispositivos automáticamente al añadirlos

RMW: Read-Modify-Write: operación que lee una ubicación de memoria y escribir un nuevo valor en forma simultánea, ya sea con un nuevo valor o alguna función del valor anterior.

RTL Register Transfer Level: Formato en el que se describe el lenguaje HDL en donde se le concede más importancia a lo que hace el sistema que a como lo hace.

SOC System on Chip: es un sistema hardware - software muy complejo que integra en un solo chip los diferentes componentes de un sistema, para poder brindar características de alto rendimiento.

Topología anillo spidergon: Topología que brinda buen compromiso en rendimiento, escalabilidad entre las arquitecturas más eficientes que existen.

VLIW Very Long Instruction Word. Es una arquitectura para diseño de CPUs que implementa una forma de paralelismo a nivel de instrucción.

BIBLIOGRAFÍA

- [1] R. Saleh, S. Wilton, S. Mirabbasi, A. Hu, M. Greenstreet y A. Ivanov, «System-on-Chip: Reuse and Integration,» *IEEE*, p. 20, 2006.
- [2] C. Metra, J. Chung Lo y F. Lombardi, «Special Section on Design and Test of Systems-on-Chip (SoC),» *IEEE*, p. 2, 2006.
- [3] J. Mollocana y J. Cadena, «DISEÑO DE HARDWARE Y SOFTWARE DE SYSTEM ON CHIP EMPLEANDO TECNOLOGIA XILINX EDK,» *ESPE*, p. 13, 2012.
- [4] W. Bram, «BLOG Bram Wolfs,» Agosto 2012. [En línea]. Available: <http://bramwolfs.com/2012/08/>.
- [5] Kong Woei Susanto, «A Verification Platform for System on Chip,» *University of Glasgow*, p. 166, 2003.
- [6] J. Brandt, K. Schneider y A. Willenbücher, «Using IP Cores in Synchronous Languages,» *University of Kaiserslautern*, p. 10.
- [7] W. Wolf, *Modern VLSI Design: IP-based design*, 4 ed., Boston: Prentice Hall, 2008, p. 627.
- [8] L. Shannon, «Impact of Intellectual Property Cores on Field Programmable Gate Array,» *University of Toronto*, p. 79, 2001.
- [9] R. Martínez y L. Terés, «Web can Nic,» [En línea]. Available: <http://www.cannic.uab.es/docencia/DCIIsII/Apunts/IPcoresIntro.pdf>.
- [10] M. J. Bellido Díaz, «Departamento de Tecnología Electrónica, Universidad de Sevilla,» 14 Octubre 2014. [En línea]. Available: <http://www.dte.us.es/docencia/etsii/gii-ic/laboratorio-de-desarrollo-hardware/temas/Tema1VGDH/view>.

- [11] «design-reuse.com,» 2009. [En línea]. Available:
<http://www.design-reuse.com/news/?id=2000&print=yes>.
- [12] M. Hübner y J. Becker, Multiprocessor System-on-Chip:
Hardware Design and Tool Integration, New York: Springer, 2011.
- [13] W. Wolf, A. Jerraya y G. Martin, «Multiprocessor System-on-
Chip (MPSoC) Technology,» *IEEE*, p. 13, Octubre 2008.
- [14] M. Reza, «Reliable and Variation-tolerant Reliable and
Variation-tolerant Power MPSoCs,» *Universita' di Bologna*, 2012.
- [15] «From “Bus” and Crossbar” to “Network-On-Chip”,» *ASTERIS
S.A*, p. 10, 2009.
- [16] «A comparison of Network-on-Chip and Busses,» *ASTERIS S.A*,
p. 11, 2005.
- [17] S. Kumar, A. Jantsch, J. P. Soininen y M. Forsell, «A Network
on Chip Architecture and Design Methodology,» *IEEE*, p. 8, 2002.
- [18] N. Roma, «Network-on-Chip for Multicore Processors,»
University of Lisbon, p. 5, 2008.
- [19] D. Atienzaa, F. Angiolinic, S. Muralia, A. Pullinid, L. Beninic y G.
D. Michelia, «Network-on-Chip design and synthesis outlook,»
Science Direct, 2007.
- [20] «University of Cagliari.,» [En línea]. Available:
<http://www.diee.unica.it/eolab2/NoC.html>.
- [21] S. Pasricha y N. Dutt, On-Chip Communication Architectures
System on Chip Interconnect, Burlington: ELSEVIER, 2008.

- [22] X. Ju y L. Yang, «NoC Research and Practice: Design and Implementation of 2x4 2D-Torus Topology,» *Information Technology and Computer Science*, p. 7, 2011.
- [23] S. S. Bhople y M. A. Gaikwad, «A Comparative Study of Different Topologies for Network-On-Chip Architecture,» *International Journal of Computer Applications* , p. 3, 2013.
- [24] A. Kaushal y S. Singh, «Network on Chip Architecture and Routing Techniques: A survey,» *International Journal of Research in Engineering and Science (IJRES)*, p. 5, 2014.
- [25] Z. Lu, «Using Wormhole Switching for Networks on Chip:Feasibility Analysis and Microarchitecture Adaptation,» *Royal Institute of Technology*, p. 70, 2005.
- [26] L. Benini y G. De Micheli, NETWORKS ON CHIPS TECHNOLOGY AND TOOLS, San Francisco: ELSEVIER, 2006.
- [27] P. Mohapatra, «Wormhole Routing Techniques for Directly Connected Multicomputer Systems,» *Lowa State University*, p. 45.
- [28] V. Rantala, T. Lehtonen y J. Plosila, «Network on Chip Routing Algorithms,» *Turku Centre for Computer Science*, p. 38, 2006.
- [29] S. Chawade, M. Gaikwad y R. Patrikar, «Design XY Routing Algorithm for Network-On-Chip Architecture,» p. 5.
- [30] S. Mubeen, «EVALUATION OF SOURCE ROUTING FOR MESH TOPOLOGY NETWORK ON CHIP PLATFORMS,» *Jönköping Institute of Technology*, p. 109, 2009.
- [31] M. Atagoziyev, «ROUTING ALGORITHMS FOR ON CHIP NETWORKS,» *MIDDLE EAST TECHNICAL UNIVERSITY*, p. 95, 2007.

- [32] Harvard University, «Benchmarking Methodology Working Group,» 1991. [En línea]. Available: <http://www.rfc-editor.org/rfc/pdf/rfc1242.txt.pdf>.
- [33] S. Bradner y W. James, «RFC 1242 Benchmarking Terminology for Network Interconnection Devices,» *IETF*, p. 12, Julio 1991.
- [34] W. Orozco y M. Aymara, «INTERCONECTIVIDAD DE ROUTERS EMULADOS MEDIANTE GNS3 CON ROUTERS EMULADOS FÍSICOS,» *ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO*, p. 149, 2013.
- [35] E. Interiano y F. M. d. Oca, «Redes de Computadoras Ethernet conmutada,» *Tecnológico de Costa Rica*, p. 68, 2012.
- [36] M. Rouse, «Round-trip time (RTT),» [En línea]. Available: <http://searchnetworking.techtarget.com/definition/round-trip-time>.
- [37] «VoipForo,» [En línea]. Available: http://www.voipforo.com/QoS/QoS_Jitter.php.
- [38] J. Vilchez, «jesusvilchez.wordpress.com,» 30 Agosto 2011. [En línea]. Available: <http://jesusvilchez.wordpress.com/2011/08/30/variacion-de-retardo-jitter/>.
- [39] D. Balasubramanian, T. Nagda, D. Bhatia, P. Sighn, S. Bhoj y P. Balsara, «Network interface for NoC based architectures,» *University of Texas and Dallas*, p. 18, 2007.
- [40] «Expertiza,» Abril 2014. [En línea]. Available: <http://wiki.expertiza.ncsu.edu/index.php/File:Spidergon.png>.
- [41] O. Salcedo, D. López y Á. Ríos, «Desempeño de la calidad del servicio (QoS) sobre IPv6,» *TECNURA*, p. 41, 2011.

- [42] J. Dielissen, A. Røadulescu, K. Goossens y E. Rijpkema, «Concepts and Implementation of the Philips Network-on-Chip,» *Philips Research Laboratories*.
- [43] A. Brinkmann , C. Niemann, I. Hehemann, D. Langen, M. Pormann y U. Ruckert, «On-Chip Interconnects for Next Generation System-on-Chips,» *IEEE*, p. 5, 2002.
- [44] Y. Jin, E. J. Kim y T. M. Pinkston, «Communication-Aware Globally-Coordinated On-Chip Networks,» *IEEE*, p. 13, 2002.
- [45] A. Ankur, I. Cyril y S. Ravi, «Survey of Network on Chip (NoC) Architectures & Contributions,» *Journal of engineering computing and architecture*, p. 15, 2009.
- [46] R. Gindin, I. Cidon y I. Keidar, «NoC-Based FPGA: Architecture and Routing,» *Israel Institute of Technology*, p. 10.
- [47] B. Grot, J. Hestness, S. W. Keckler y O. Mutlu, «A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees,» *International Symposium on Computer Architecture*, p. 12, 2011.
- [48] J. Martinez, «Calidad de Servicio QoS,» *Pontificia Universidad Javeriana de Cali*, p. 70.
- [49] «Calidad de Servicio y Manejo del Ancho de banda,» 2008. [En línea]. Available: <http://www.mkresolutions.net/soluciones/calidad-de-servicio-qos-ancho-de-banda/>.
- [50] J. Gómez, «Calidad de Servicio (QoS),» *adminso.es*, p. 94.
- [51] «Open Core Protocol,» *OCP-IP Association*, p. 2, 2004.

- [52] B. Attia, W. Chouchene, A. Zitouni y R. Tourki, «Network Interface Sharing for SoCs based NoC,» *IEEE*, p. 6, 2011.
- [53] LordT, «Wikimedia,» Diciembre 2007. [En línea]. Available: http://es.wikipedia.org/wiki/Modelo_OSI#mediaviewer/File:Pila-osi-es.svg.
- [54] J. L. Sánchez García, «Análisis y evaluación de técnicas de reconfiguración dinámica de la red de interconexión en sistemas masivamente paralelos,» *Universidad de Castilla-La Mancha*, p. 245, 1998.
- [55] A. K. Shrimawale y M. A. Gaikwad, «Review of Odd Even Routing Algorithm for 2D Mesh Topology of Network-on-Chip Architecture for Bursty Traffic,» *International Journal of Computer Applications*, p. 12, 2013.
- [56] A. K. Shrimawale y M. A. Gaikwad, «Review of Odd Even Routing Algorithm for 2D Mesh Topology of Network-on-Chip Architecture for Bursty Traffic,» *International Journal of Computer Applications*, p. 12, 2013.
- [57] J. Lee, S. J. Park, M. Swaminathan y J. Kim, «Do We Need Wide Flits in Networks-On-Chip?,» *University of Cyprus*, p. 6.
- [58] XILINX, «Virtex-6 CXT Family Data Sheet,» *XILINX*, p. 52, 2011.
- [59] L. A. Guijarro Coloma, «Contribución al soporte de la clase de servicio ABR en redes ATM mediante técnicas de planificación equitativa,» *Universidad Politécnica de Valencia*, p. 205, 1998.
- [60] J. E. Rivas Ramírez, «ANÁLISIS DE AUTOSIMILARIDAD PARA FUENTES SINTÉTICAS DE VIDEO,» *UNIVERSIDAD CATÓLICA DE MANIZALES*, p. 74, 2013.

- [61] M. A. Castro Gil y A. Comenar Santos, «Tecnología Multimedia-Video Digital,» *Universidad Nacional de Educación a Distancia*, p. 116.
- [62] E. Casilari Pérez, «CARACTERIZACIÓN Y MODELADO DE TRÁFICO DE VÍDEO VBR,» *UNIVERSIDAD DE MÁLAGA*, p. 378, 1998.
- [63] F. Fitzek y M. Reisslein , «Telecommunication Networks Group (TKN),» 02 2003. [En línea]. Available: <http://www-tnk.ee.tu-berlin.de/research/trace/stat.html>.
- [64] L. Jain, «NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling,» *NIRGAM Authors*, p. 28, 2007.
- [65] M. Rouse, «WhatIs,» Marzo 2011. [En línea]. Available: <http://whatis.techtarget.com/definition/IP-core-intellectual-property-core>.
- [66] «ItWisen,» [En línea]. Available: <http://www.itwissen.info/definition/lexikon/System-on-Chip-SoC-system-on-chip.html>.
- [67] O. Salcedo, «Desempeño de la Calidad de Servicio (QoS)sobre Ipv6,» *Tecnura*.
- [68] S. Bradner, «The Internet Engineering Task Force,» JULIO 1991. [En línea]. Available: <https://www.ietf.org/rfc/rfc1242.txt>.
- [69] W. Dally y B. Towles, *PRINCIPLES AND PRACTICES OF INTERCONNECTION NETWORKS*, Elsevier, 2004.
- [70] C. Timothy y G. Seth, «NIFDY: A Low Overhead, High Throughput Network Interface,» *22nd Annual International Symposium on*, p. 12, 1995.

- [71] K. Swaminathan, G. Lakshminarayanan, F. Lang, . M. Fahmi y S. Bum Ko, «DESIGN OF A LOW POWER NETWORK INTERFACE FOR NETWORK ON CHIP,» *IEEE*, p. 4, 2013.
- [72] T. M. A. Y. Nizar Dahir, «Plane-Balanced and Deadlock-Free Adaptive Routing for 3D Networks-on-Chip,» *NewCastle University*, p. 21, 2012.
- [73] Xilinx, «ML605 Hardware User Guide,» *Xilinx*, p. 94, 2012.
- [74] L. Jain, «NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling,» *University of Southampton*, p. 34, 2007.