République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université SAAD DAHLEB – Blida 1
Faculté des Sciences
Département d'Informatique

PROJET DE FIN D'ETUDES

Pour obtenir le titre de master en Informatique
Option : Système d'information et réseaux

Sujet

Modélisation et validation d'une architecture d'un réseau sur puce NOC

Réalisé par :
KOUIDER ELOUAHED Elferdaous
OUADFEL Nour el houda

Devant le jury composé de :

Mr. Ould khaoua Mohammed, Département d'Informatique, Blida 1	**Président**
Mme. Laaroussi Sana, Département d'Informatique, Blida 1	**Examinatrice**
Mme.Daoud Hayat, Département d'Informatique, Blida 1	**Promotrice**

Promotion : 2021/2022
Session : juillet 2022

Résumé

La technologie est devenue une partie intégrante dans nos vies quotidiennes grâce à l'évolution des systèmes sur puce. Ces systèmes sont des circuits intégrés dans une seule puce.

La communication entre les composants du SOC est faite par des différents types d'interconnexion (point à point, bus partagé et bus hiérarchique) par contre ces dernières rencontres des inconvénients. C'est pour cela, ils ont évolué un nouveau type qui s'appelle les réseaux sur puce (Network On Chip).

Les réseaux sur puce sont divisés sur plusieurs types, dans notre projet on s'intéresse sur les QNOC (réseau sur puce avec une qualité de service) d'une part, et sur l'algorithme de routage XY adaptatif qui achemine les paquets entre les routeurs d'autre part.

Dans notre travail, nous avons étudié le problème d'inter-blocage qui se pose dans l'algorithme de routage XY adaptatif et nous avons ajouté une condition dans cet algorithme de telle manière que les problèmes soient réglés.

Mots clé : système sur puce, réseau sur puce, réseau sur puce avec une qualité de service (QNOC), algorithme de routage XY adaptatif.

Abstract

Technology has become an integral part of our daily lives with the evolution of system-on-chip. These systems are integrated circuits in a single chip.

The communication between the components of the SOC is made by different types of interconnection (point to point, shared bus and hierarchical bus) on the other hand these last encounters disadvantages. This is why they have evolved a new type called Network On Chip.

Networks on chip are divided on several types, in our project we are interested in QNOC (network on chip with quality of service) on the one hand, and on the adaptive XY routing algorithm which routes packets between routers on the other hand.

In our work, we have studied the deadlock problem that arises in the adaptive XY routing algorithm and we have added a condition in this algorithm in such a way that the problems are solved.

Keywords: system-on-a-chip, network-on-a-chip, network-on-a-chip with quality of service (QNOC), adaptive XY routing algorithm.

ملخص

أصبحت التكنولوجيا جزءا لا يتجزأ من حياتنا اليومية بفضل تطور النظام على شريحة، وهذه الأنظمة هي دوائر متكاملة بواسطة أنواع مختلفة من الربط البيني (من نقطة إلى نقطة، حافلة SOC يتم الاتصال بين مكونات .في شريحة واحدة مشتركة وحافلة هرمية) من ناحية أخرى، يواجه الأخير عيوبا وهذا هو السبب في أنهم طوروا نوعا جديدا يسمى الشبكات تنقسم الشبكات على الشريحة إلى عدة أنواع، في مشروعنا نحن مهتمون ب .(Network On Chip) على الشريحة التكيفية التي توجه الحزم بين XY (شبكة على شريحة ذات جودة خدمة) من ناحية، وعلى خوارزمية توجيه QNOCS وافترضنا حلا بطريقة يتم XY في عملنا درسنا المشكلة التي تنشأ في خوارزمية توجيه .أجهزة التوجيه من ناحية أخرى حلها.

الكلمات المفتاحية : نظام على رقاقة، شبكة على رقاقة، شبكة على رقاقة مع جودة الخدمة، خوارزمية التوجيه التكيفية

Remerciements

Ce travail est l'aboutissement d'un dur labeur et de beaucoup de sacrifices ; nos remerciements vont d'abord au Créateur ALLAH de l'univers qui nous a doté d'intelligence, et nous a gardées en bonne santé.

Tout d'abord, nous tenons à remercier notre promotrice Mme DAOUD Hayet. Les mérites du mémoire appartiennent certes à l'auteur, mais aussi au Directeur qui le supervise. Dans notre cas, notre Directrice a été d'un soutien et d'une attention exceptionnelle. La confiance qu'elle nous a confronté aux problèmes et d'essayer de nous guider pour trouver des solutions.

Nous tenons à remercier les jurys d'accepter de juger notre travail.

Nous souhaitons aussi exprimer nos profondes gratitudes au Chef de Département d'Informatique Mrs Ahmed OULD AISSA, et l'ensemble des enseignants et personnels au sein de notre Département.

Nous ne pouvons passer outre notre reconnaissance envers nos familles lesquelles Nous disons tout simplement MERCI. Leurs présences, leurs écoutes, leurs confiances et leurs soutiens constants nous a assuré des bases solides qui nous permettent de persévérer et de nous surpasser.

Enfin, Nous remercions tous ceux qui nous ont aidées de près ou de loin dans l'élaboration de ce travail.

Dédicace

C'est avec profonde gratitude et sincères mots, que Je dédie ce modeste travail accompagné d'un immense amour

À mon cher père, Qui m'a soutenu depuis le début et qui m'a fait confiance, tout mon amour et ma reconnaissance à lui.

À ma chère mère pour tous ses sacrifices, son amour, sa tendresse, son soutien et ses prières tout au long de nos études.

À mes sœurs Ilhem, Ferial et aya.

À mes frère Zaki et Iyad.

À mes amies merouane, baya, hana, samah, wafa.

À ma binôme Elferdaous.

Nour el houda

Dédicace

J'ai le grand plaisir de dédié ce modeste travail à :

Mes très chers parents qui m'ont guidée durant les moments les plus pénibles de ce long chemin, ma mère qui a été à mes coté et ma soutenue durant toute ma vie, et mon père qui sacrifié toute sa vie afin de me voir devenir ce que je suis, merci mes parents.

A mes sœurs : Menel, Asma, Khaoula et Romeissa

A mes petites nièces : Lina, Maria, Elya, et Sidra

A mes petits neveux : Amir, Wassim, Yacine, Youcef et Younes

A tous mes amis surtout Manel, Hana, Wafa et riyadh

Enfin je remercie ma binôme Nour el houda qu'elle a contribué à la réalisation de ce modeste travail.

Elferdaous

Table des matières

Liste des figures

Liste de tableau

Liste des abbreviations

Abréviation	Signification
ACK	ACKnowledgement « accusé de réception »
ALG	Asynchronous Latency Guarantee
ASIC	Application Specific Integrated Circuit
ASPIN	Asynchronous Scalable Predictable Interconnect Network
BFT	Byzantine Fault Tolerance
CHAIN	CHip Area INterconnect
CI	Circuit Intégré
CLB	Configurable Logic Bloc
CLK	CLocK
CLP	Composant logique programmable
CMOS	Complementary-symmetry/Metal-Oxide Semiconductor
CPLD	Complex Programmable Logic Device
CU	Central Unit
DMA	Direct Memory Access
DSP	Digital signal processor
DSPIN	Distributed Scalable Predictable Interconnect Network
DYXY	DYnamic XY
EPLD	Erasable Programmable Logic Device
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In First Out
FLIT	FLow control unIT
FPGA	Field Programmable Gate Arrays ou "réseaux logiques programmables
FSM	Finite State Machine
IOB	Input Output Bloc
IP	Intellectual Property
LUT	Look-Up Table
NA	Network Adaptateur
NI	Network Interface
NOC	Network On Chip
OCP	Open Core Protocol
PAL	Programmable Array Logic
PCB	Printed Circuit Board
PDA	Personal Digital Assistants
PE	Process Element
PIR	Packet Injection Rate
PLA	Programmable Logic Array
PLD	Programmable Logic Device
PLL	Phase-Locked Loop
PROM	Programmable Read Only Memory
QNOC	Quality-of-Service Network On Chip
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register Transfer Logic
SL	Service level

Liste des abbreviations

SOC	Systéme On Chip
SPIN	Scalable Programmable Integrated Network
SRAM	Static Random Access Memory
TDM	TimeDivisionmultiplexing
TG	générateur de trafic
TRA	Time-routing
TTM	Time To Market
VC	Virtual Circuit
VCT	Virtual Cut Through
VHDL	(Very High Speed Integrated Circuit) Hardware Description Language

INTRODUCTION GENERALE

Avant les années 2000, la communication dans les circuits intégrés sur la même puce (Système sur puce) a été basée sur les types d'interconnexion classique (point à point, bus et bus hiérarchique).

Après l'évolution de la technologie, les systèmes embarqués sont grandis de plus en plus et leurs besoins ont grandi aussi, donc les systèmes sur puce implémenté un nombre croissant d'unité de traitement et la quantité des données transmis entre ces unités augmente et les paradigmes d'interconnexion connaissent des limites dans la bande passante du bus, difficulté d'adaptation des débits des échanges aux besoins des applications, latences importantes, et impossibilité de bien gérer le parallélisme dans le traitement des données compte tenu des contraintes d'acheminement des informations et des éléments de contrôle[1].

C'est pour cela les chercheurs ont développé un nouveau type d'interconnexion qui est les réseaux sur puce (NOC).

Le réseau sur puce est un schéma qui organise la communication entre les différents modules sur la même puce, il permet de disposer d'architectures de communication flexibles et performantes et de supporter un nombre grandissant de processeurs de calcul. L'élément de base dans cette topologie est le routeur, qui est responsable sur l'acheminement des paquets dans la zone active reconfigurable, il utilise pour cela des algorithmes de routage, parmi ces algorithmes on est intéressé par l'algorithme XY tolérant aux fautes, Il est appelé adaptatif ou tolérant aux fautes car il permet de calculer dynamiquement le chemin emprunté par message. Mais cela peut rencontrer des problèmes principaux : les erreurs aux échecs de décisions de routage et le problème des nœuds défectueux rencontré en chemin qui nous causent la défaillance du matériel.

C'est pour cela, nous avons étudié dans ce travail l'algorithme XY adaptatif et nous avons proposé un algorithme de routage adaptatif qui permet de tester la performance et détecter les limites des switchs du QNOC tolérant aux fautes, sachant que l'algorithme proposé par le laboratoire LCOMS est performant sauf certaines situations ne sont pas débloqués, alors on propose une amélioration de cet algorithme en débloquant ces situations. L'ajout de certaines conditions dans l'algorithme permet d'enlever ce problème en voyant le résultat dans la simulation.

Pour mener le bon déroulement de notre travail, nous l'avons divisé en quatre chapitres tel que le premier chapitre rappelle tout d'abord, la présentation des différentes définitions et d'une façon générale les systèmes sur puce

Le deuxième chapitre est consacré pour un type de NOC qui est le QNOC.

Dans le troisième chapitre nous parlerons sur le problématique et la solution proposée.

Enfin dans le quatrième chapitre, nous allons faire l'implémentation, les tests nécessaires et nous terminerons par les analyses des résultats obtenues.

Ce mémoire sera finalisé par une conclusion générale et quelques perspectives de notre travail.

CHAPITRE I : Généralités sur les réseaux sur puce

I.1. Introduction

Le monde a connu une terrible évolution de la technologie. Les systèmes embarqués sont évolués, l'homme est devenu indispensable de cette dernière dans sa vie quotidienne d'où la plus grande partie des systèmes informatiques utilisés dans nos jours sont des systèmes embarqués tels que les téléphones portables, les satellites artificiels, le réfrigérateur etc...

Toutes ces machines sont composées de plusieurs composants (mémoire, cpu, transistor...) sur une seule puce, ce système est appelé système sur puce (**Systèm On Chip**).

La communication entre ces dernières influences sur la performance du système, à l'époque ils ont utilisé des paradigmes d'interconnexion comme point à point, bus d'hiérarchique et bus partagé mais ces technologies ont connu plusieurs problèmes, c'est pour cela ils ont créé un nouveau paradigme appelé réseau sur puce (**Network On Chip**). Ce dernier a plusieurs avantages par rapport aux autres interconnexions. Parmi ces avantages une bonne flexibilité de communication et une consommation réduite d'énergie et de surface.

Pour aborder le travail qui va suivre, il est nécessaire de définir les concepts utiles à la compréhension et l'appréhension de ce projet. Nous allons présenter un aperçu sur les systèmes embarqués, par la suite on va définir les systèmes sur puce et les types d'interconnexion. On va expliquer aussi les circuits FPGA et ASIC, A la fin on va voir les concepts des réseaux sur puce et leurs différentes topologies et leurs types.

I.2.Systèmes Embarques

I.2.1. Définition

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires [2]. Un système embarqué est autonome et ne possède pas des entrées/sorties standards.

I.3.Système sur puce

I.3.1. Définition

Système sur une puce ou bien un ***system on chip*** (SOC) est un circuit intégré qui rassemble les principaux composants d'un ordinateur sur une seule puce électronique[3], pouvant comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la fonction attendue.

I.3.2. Différentes technologies de système sur puce

Les circuits intégrés les plus simples sont des portes logiques (et, ou, non), les plus complexes sont les microprocesseurs et les plus denses sont les mémoires. On trouve de nombreux circuits intégrés dédiés à des applications spécifiques (ASIC pour Application Specific Integrated Circuit), notamment pour le traitement du signal (traitement d'image, compression vidéo...) on parle alors de DSP (pour Digital Signal Processor). Une famille importante de circuits intégrés est celle des composants de logique programmable (FPGA, CPLD)[4].

I.3.2.1. FPGA

I.3.2.1.1 Définition

FPGA (Field Programmable Gate Arrays ou "réseaux logiques programmables"), c'est un circuit intégré fait pour être reprogrammé par l'utilisateur après sa fabrication en utilisant un langage informatique spécifique sans modifier le matériel.

Les circuits FPGA ressemblent aux puces de mémoire morte programmable (PROM), mais leur potentiel d'application est bien plus vaste.

Les FPGA contiennent des blocs logiques programmables pouvant être câblés dans différentes configurations. Ces blocs créent un tableau physique des portes logiques qui peut être utilisé pour effectuer différentes opérations.

À terme, les circuits FPGA pourraient permettre aux utilisateurs de fabriquer des microprocesseurs adaptés à leur besoins propres.

I.3.2.1.2 Application des FPGA

Les FPGA sont utilisés dans de nombreuses applications [5] par exemple :

- Prototypage de nouveaux circuits.
- Fabrication de composants spéciaux en petite série

- Adaptation aux besoins rencontrés lors de l'utilisation.
- Systèmes de commande à temps réel.
- DSP (Digital Signal Processor).
- Imagerie médicale.

I.3.2.1.3 Architecture des FPGA

La couche dite 'circuit configurable' est constituée d'une matrice de blocs logiques configurables CLB permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Ces blocs sont entourés par des blocs entrées/sorties IOB qui gèrent les entrées-sorties réalisant l'interface avec les modules extérieurs, l'ensemble est relié par un réseau d'interconnexions programmable (figure I.1). [6]

Il y a 4 principales catégories disponibles commercialement [6]:

- Tableau symétrique.
- En colonne.
- Mers de portes.
- Les PLD hiérarchique.

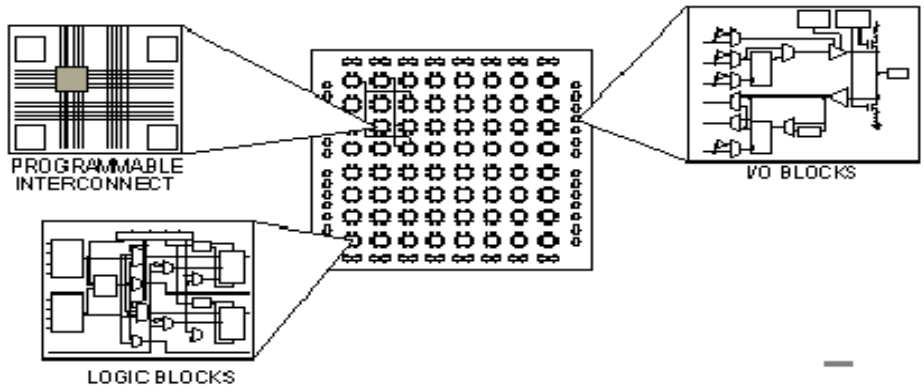


Figure I.1: architecture interne d'un FPGA[6].

Les circuits FPGA du fabricant Xilinx utilisent deux types de cellules de base :

- Les cellules d'entrées/sorties appelés IOB (input output bloc),
- Les cellules logiques appelées CLB (configurable logic bloc). Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable [6].

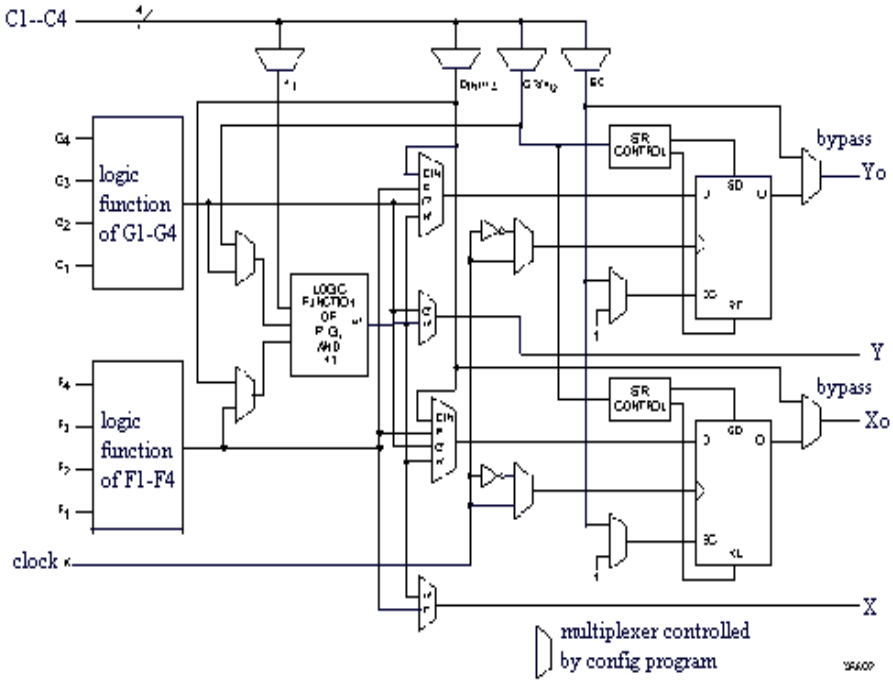


Figure I.2 : cellule logique (CLB)[6].

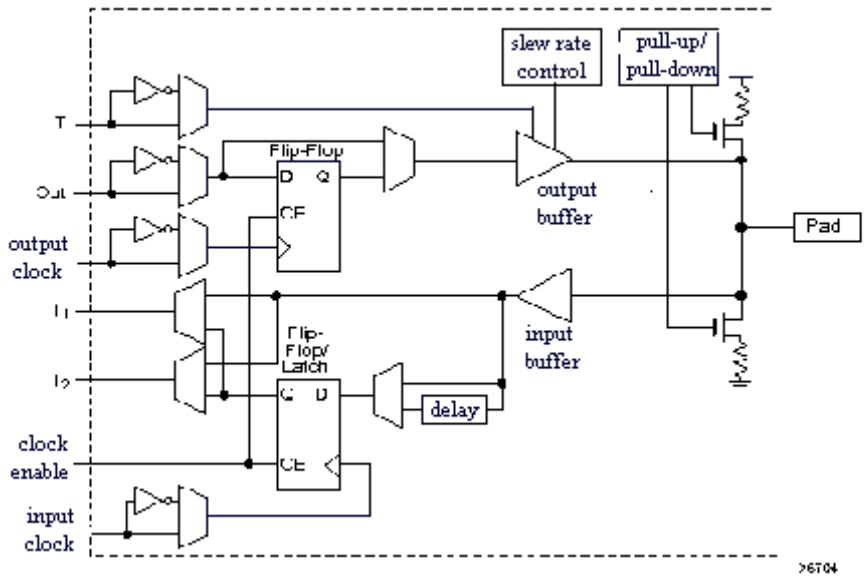


Figure I.3: input output bloc[6].

I.3.2.1.4. Les types de blocs logiques dans les FPGA

Il existe quatre types de blocs logiques :

1. **Cellules symétriques :** Ces blocs de petite taille servent à réaliser des fonctions logiques simples mais aussi comme ressources de connexion [7].

2. **Les Multiplexeurs** : En venant programmer les entrées du multiplexeur et en pilotant les signaux de sélection, il est possible de réaliser toutes les fonctions logiques (autant d'entrées que de signaux de sélection). Leur très petite taille est particulièrement adaptée à la technologie anti-fusible [7].

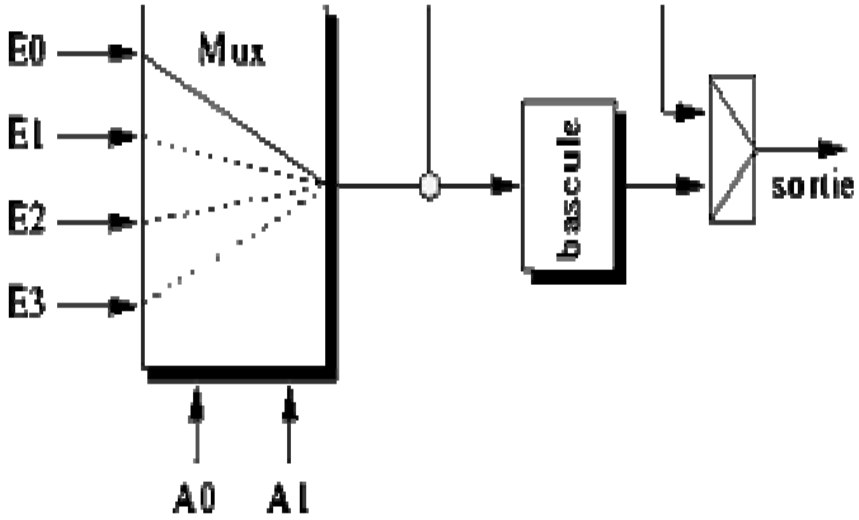


Figure I.4: multiplexeur [8].

3. **Les LUT (Look-Up Table):** Les Look-Up Table sont un cas particulier des cellules à multiplexeur, avec de 2 à 9 entrées et dont la technologie du point mémoire est une technologie SRAM. Une Look Up Table de N entrées est une mémoire qui peut implémenter n'importe quelle fonction booléenne de N variables [8].

Les N entrées sont utilisées comme adresse d'une mémoire de 2n bits qui code la fonction booléenne à réaliser. On peut donc réaliser 2n fonctions différentes avec une LUT à N entrées. Les Look-Up Tables sont des blocs logiques de très petite granularité dans un CLP. Comme Les Look-Up Table possèdent une bascule de sortie, les architectures à base de Look-Up Table sont beaucoup plus riches en bascules que les architectures à macro-cellules [8].

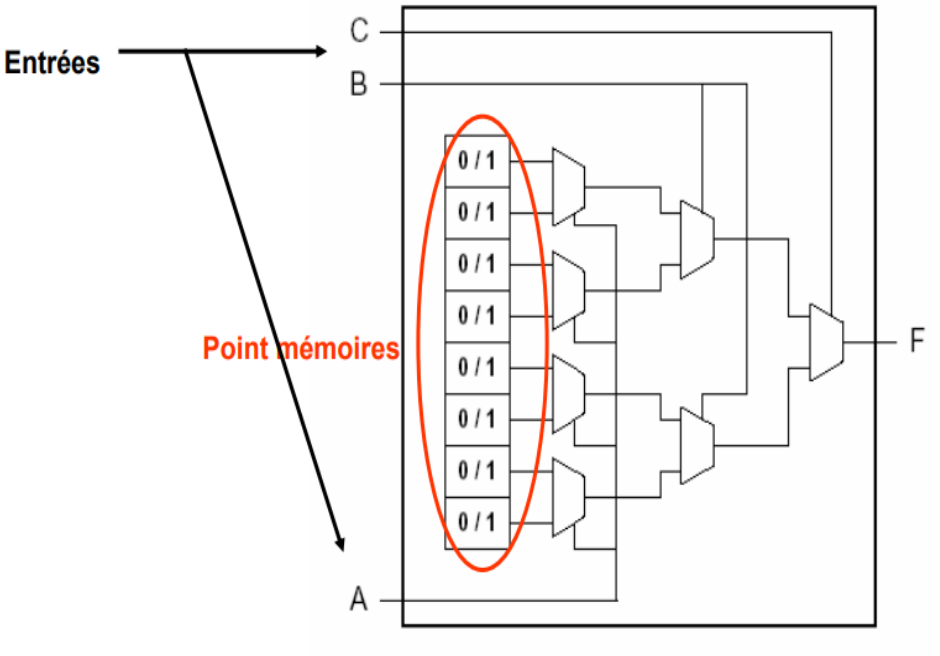


Figure I.5: table de look-up [8].

4. **Les Macro-cellules :** On retrouve dans certaines macro-cellules toute la complexité d'un PAL. Il y a en général un nombre réduit de macrocellules dans un EPLD, car ces cellules occupent une grande surface [7].

La macro-cellule présente plusieurs intérêts :

- Grand nombre de variables d'entrées possibles [8].
- Grand nombre de termes de produits possibles [8].

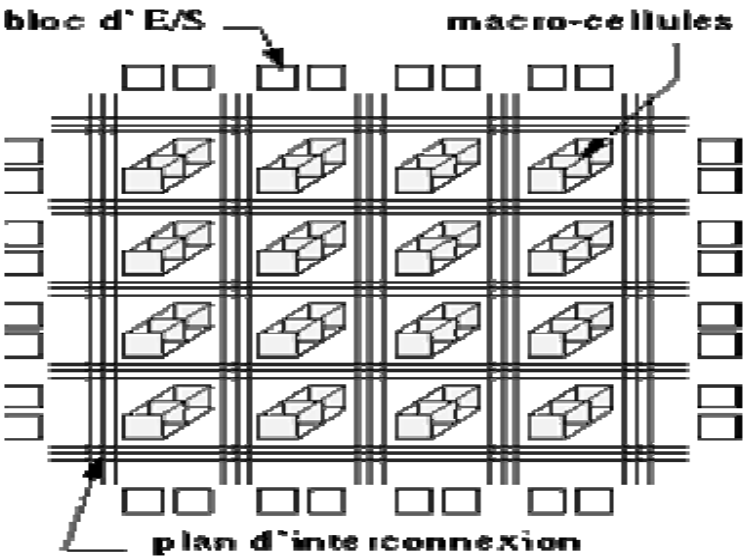


Figure I.6: macro-cellules [8].

I.3.2.2. ASIC

I.3.2.2.1. Définition

Un ASIC, (***Application Specific Integrated Circuit)*** en anglais, ou (***Circuit intégré propre à une application)***, en français, est un type de circuit électronique particulier. Autrement dit, l'ASIC est un circuit intégré exclusivement dédié à une application et à un utilisateur.

Les circuits ASIC sont utilisés dans de nombreuses applications, comme le contrôle des émissions des véhicules, la surveillance de l'environnement et les assistants numériques personnels (PDA, Personal Digital Assistants).

I.3.2.2.2. Les types d' ASIC

L'histoire des conceptions et de la technologie ASIC peut être caractérisée par la croissance continue et l'évolution des différents styles de conception de l'ASIC. Statistiquement parlant, les ASIC de type CMOS est le type dominant, mais il existe plusieurs autres types de modèles ASIC.

En gros, tous les ASIC peuvent être classés en trois catégories sur-mesure (full-custom), le moitié sur-mesure (semi-custom) et le programmable.

La figure suivante montre les différents types d'ASIC ainsi que les sous-catégories de chaque type.

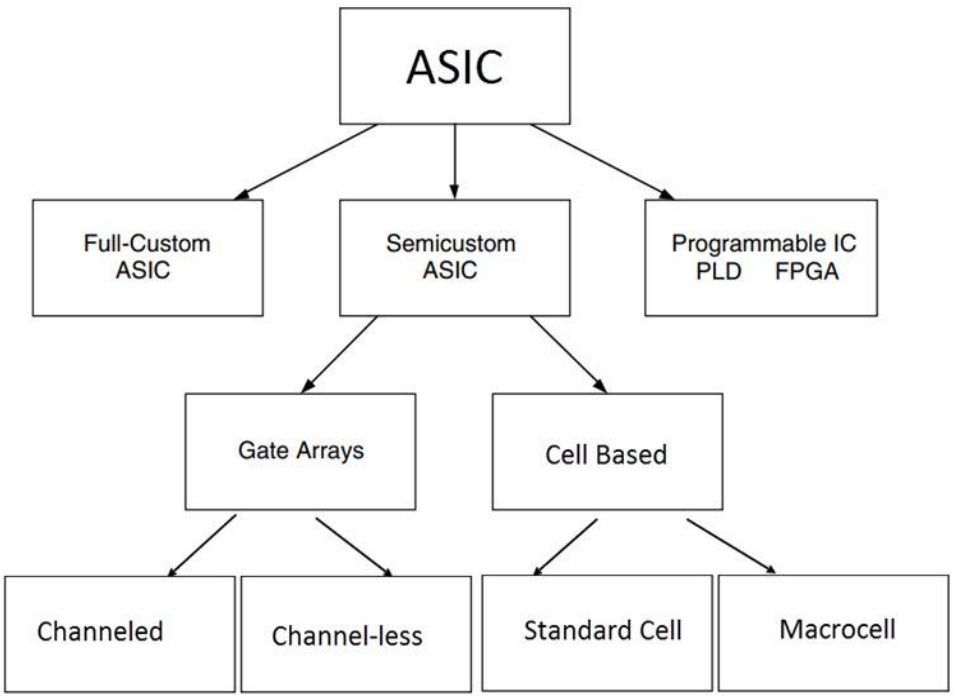


Figure I.7: les types d'ASIC [9].

1- Sur mesure *(full-custom)* :

Dans Full-Custom ASIC, toutes les cellules logiques, les circuits et les dispositions sont conçus spécifiquement pour cet ASIC particulier à partir du sol. Le concepteur peut choisir une conception ASIC complète uniquement s'il pense que les bibliothèques existantes ne sont pas assez rapides ou que les cellules logiques ne sont pas petites ou que la consommation d'énergie est élevée.

Les principaux avantages des ASIC sur mesure par rapport aux autres modèles de CI sont qu'ils offrent les meilleures performances possibles à la plus petite taille de matrice possible. Mais cette performance élevée et cette petite taille se traduisent par une augmentation du temps de conception, une conception complexe et le coût global du CI lui-même [9].

Les ASIS complets les plus courants sont les microprocesseurs, les mémoires, les processeurs analogiques, les dispositifs de communication analogiques/numériques, les capteurs, les transducteurs, les circuits intégrés haute tension pour automobiles, etc.

2-* Moitié sur-mesure *(semi-custom) :

Pour raccourcir le temps de conception et réduire le coût des ASIC sur mesure, de nombreuses autres approches de conception ont été développées et celles-ci sont appelées ASIC moitié sur-mesure. Habituellement, le niveau le plus bas de la hiérarchie impliquée dans la conception moitié sur-mesure est le niveau logique ou niveau de porte. Cela contraste avec le travail sur mesure, où la conception et la disposition du transistor individuel pourrait être impliqué.

Comme mentionné précédemment, la conception moitié sur-mesure de l'ASIC peut être divisée en Gate Arrays et Standard Cells. Voyons un peu de ces types [10].

- **Réseau de portes ASIC**

Dans les ASIC basés sur Réseau de portes (Gate Array), les transistors de type PN sont prédéfinis sur une plaquette de silicium en tant que tableaux. Sur la base de la conception du client et des interconnexions obtenues à partir de la conception, le fournisseur de silicium fournit ces wafers de base. Par conséquent, la plaquette de base est spécifique au client car elle est conçue sur la base des connexions fournies par le client entre les transistors du réseau de portes.

Les réseaux de portes sont à nouveau divisés en deux types appelés le réseau de portes canalisées et le réseau de portes sans canal. Dans les réseaux de portes canalisées, les interconnexions entre les cellules logiques sont effectuées dans les canaux prédéfinis entre les

lignes des cellules logiques. Dans le cas de grilles sans canal, les connexions sont faites sur une couche métallique supérieure au-dessus des cellules logiques [10].

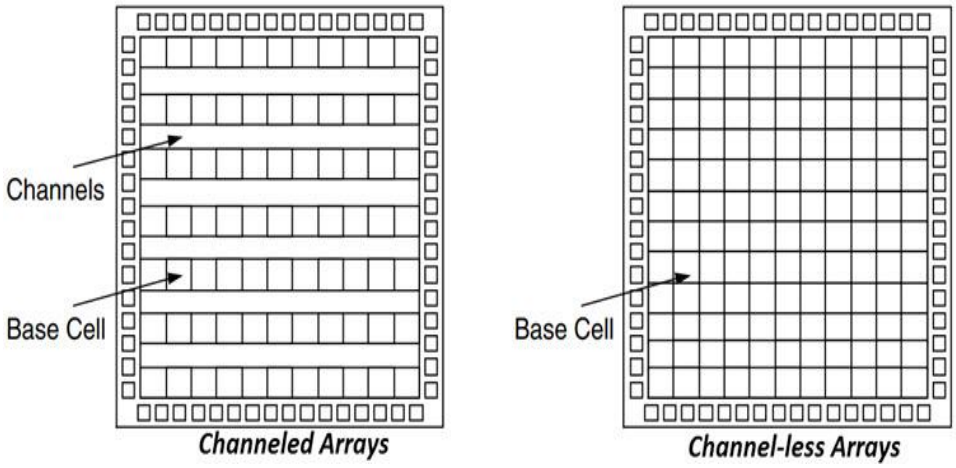


Figure I.8: les types de réseau de portes ASIC[10].

- **Les ASICs à base de cellules standards :**

Les ASIC à base de cellules standards sont les circuits les plus populaires dans la catégorie des circuits numériques car ils offrent les meilleures performances (vitesse, puissance, surface) tout en conservant un coût attractif (pour une capacité élevée) et un excellent TTM. L'abondance des cellules utilisées dépend de la technique et du fournisseur de la technique. L'architecture est constituée de parties actives basées sur deux types de cellules. Les cellules standards telles que les portes combinatoires et les portes séquentielles (INV, NAND, OR, bascule D, bascules, etc.) et les macros qui utilisent des éléments spéciaux tels que la mémoire (RAM, ROM, ...).Ou un générateur d'horloge tel qu'un PLL.

Cette architecture comprend également des cellules d'E/S (IO pads). Vous pouvez distinguer les deux types d'E/S. IO dédié à l'alimentation du circuit et IO dédié au transfert de données. La taille des IO est généralement supérieure, de l'ordre de 50 µm x 100 µm, cette taille facilite le packaging.

3- Programmable :

Pour les circuits intégrés programmables, tous les dispositifs logiques programmables tels que PAL, PLA, PLD basé sur EPROM (EPLD), PLD basé sur EEPROM (EEPLD) et les dispositifs programmables sur le terrain tels que les FPGA entrent dans cette catégorie [10].

I.3.2.3. Différence entre FPGA et ASIC

Les deux technologies présentées précédemment ont pour but la réalisation d'une fonctionnalité électronique (traitement, communication, ...) via un circuit intégré, mais le choix d'utiliser la technologie FPGA ou la technologie ASIC dépend des besoins et des objectifs (notamment en termes de performances, de coût et de TTM) de l'utilisateur. La comparaison entre les deux cibles peut se faire en trois points : performances, coût et flexibilité :

- **La performance :** Les trois principaux paramètres de performance d'un circuit intégré sont la vitesse, la surface et la consommation.

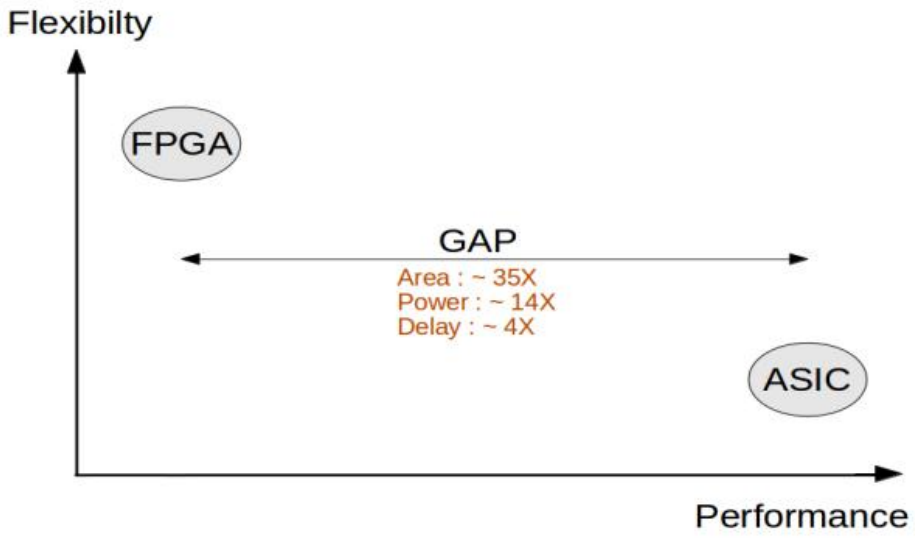


Figure I.9: l'écart de performance entre les FPGA et les ASIC [11].

En termes de vitesse, une implantation sur FPGA est entre 3,4 à 4,6 fois plus lente, en moyenne, qu'une implantation sur ASIC à base de cellules standard [12] à technologie identique. Pour ce qui est de la consommation : En moyenne, un FPGA consomme 14 fois plus qu'un ASIC (consommation dynamique) [11]. La surface pour une implantation sur FPGA en utilisant des LUT comme éléments logiques de base est d'environ 35 fois plus grande qu'une implantation sur ASIC à base de cellules standard.

- **Le coût :**

La conception d'un système sur ASIC est très coûteuse, comparée à celle sur un FPGA pour de faibles volumes à cause du coût du "ticket d'entrée" (coût fixe). Ce handicap peut être évité si la quantité de production est élevée.

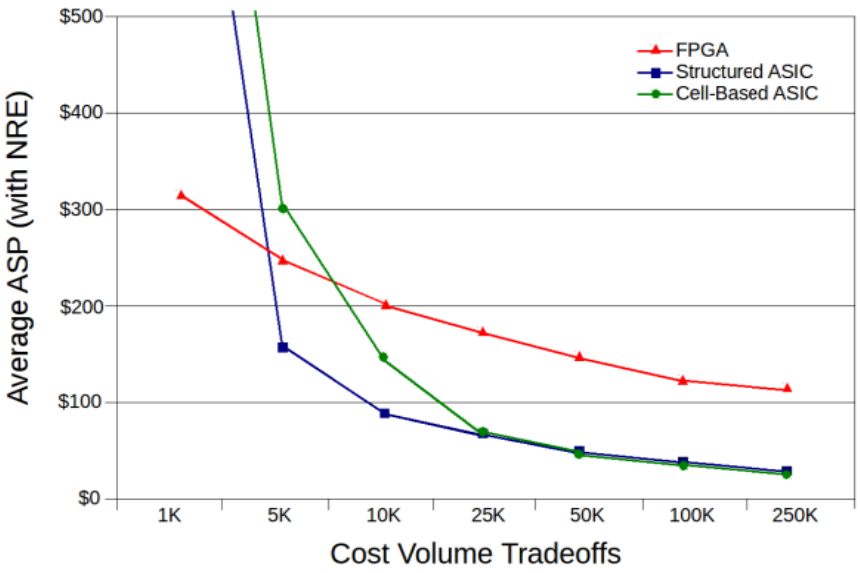


Figure I.10: Coût par rapport au volume de production [12].

- **La flexibilité :**

Le FPGA offre une grande souplesse de configurables d'un point de vue matérielle. L'ASIC par contre ne l'est pas : une fois le circuit fabriqué, l'architecture est figée et la seule fonctionnalité qu'il peut réaliser et celle décrite par la spécification mis en place des solutions de traitement pour la complexité croissante des applications [13].

I.3.3. Avantages et inconvénients du système sur puce

- **Les avantages** [14].

Taille :-Tous les composants sont maintenant sur la même puce, donc un gain de place important.

Vitesse : -Les distances entre les composants sont réduites, donc la fréquence d'horloge peut augmenter considérablement.

-Les composants sur puce et leur connexion sont plus facilement prévisibles, moins de marge.

Cout : -Un circuit imprimé moins complexe.

-Moins de soudage.

-Fréquence d'horloge faible sur la carte mais importante sur puce, cela réduit les frais liés à PCB.

Les inconvénients

Le seul inconvénient est qu'il n'autorise aucune mise à jour du matériel, si on veut faire une mise à jour il faut qu'on la faire pour chaque composant.

I.3.4. Evaluation de système sur puce

Avant les années 2000, les anciens paradigmes d'interconnexions sont utilisés dans les systèmes sur puce comme point à point, bus partagé et bus hiérarchique. Apres l'évolution du système sur puce et de la technologie ils ont créé une nouvelle technologie d'interconnexion pour éviter les problèmes des anciens paradigmes qui s'appelle réseau sur puce « **Network On Chip** ».

	parallélisme	**consommation**	**Mise en échelle**	**Réutilisation**
Point à point	Très bon	Bon	Très mauvais	Très mauvais
Bus partagé	Très mauvais	Très mauvais	Très mauvais	Très bon
Bus hiérarchique	Bon	Mauvais	Mauvais	Très bon
Réseau sur puce	Très bon	Très bon	Très bon	Très bon

Tableau I.1: la comparaison entre les différentes interconnexions [15].

I.4.Réseau sur puce

I.4.1. Définition

Le réseau sur puce (NOC) **Network On Chip** est un schéma d'organisation de la communication entre les modules d'exploitation situés sur la même puce. Il vise à combiner des cœurs de calcul à des fins diverses (exécutif, graphique, physique, etc.), des contrôleurs de périphériques, des modules ROM et RAM, des périphériques autonomes, des capteurs et bien plus encore pouvant être placés sur des cristaux de silicium [16].

I.4.2. Les caractéristiques de Réseau sur puce

Le réseau sur une puce est conçu de telle manière que les messages peuvent circuler du module source vers le module de destination via plusieurs liaisons qui impliquent des décisions de routage au niveau des commutateurs. Il dispose de plusieurs liaisons de données point à point qui sont interconnectées par des commutateurs. Il peut être classé comme un réseau de matrice commutée homogène [16].

Le réseau sur une puce présente les caractéristiques suivantes [16] :

- Le réseau sur puce permet de simplifier le matériel requis pour les fonctions de routage et de commutation.
- La prise en charge de plusieurs topologies et options multiples est possible pour différentes zones du réseau.
- L'évolutivité, l'interopérabilité et le développement de fonctionnalités sont améliorés lorsqu'ils sont combinés avec un réseau sur une puce.
- L'efficacité énergétique des systèmes sur puces complexes est améliorée avec le réseau sur puce par rapport à d'autres conceptions.
- Les problèmes de synchronisation sont mieux traités que dans d'autres conceptions. La congestion du routage des câbles présente dans la plupart des systèmes sur puces est également mieux gérée par le réseau sur une puce.
- Le réseau sur puce fournit des fréquences de fonctionnement plus élevées.
- La fermeture du calendrier est beaucoup plus facile à mettre en œuvre.
- La vérification des problèmes est beaucoup plus facile, grâce à son approche bien conçue et en couches.

I.4.3. Les composants de base du Réseau sur puce

Un réseau sur puce est base sur trois composants :

- **Les adaptateurs réseau (NA) :** Ils réalisent l'interface entre le protocole du NOC et celui des blocs IP qui composent le système. Leur rôle est de séparer le traitement (effectué dans les IP) des communications (acheminées par le réseau). Un adaptateur réseau peut être séparé en deux parties, l'interface réseau (NI) et l'adaptateur de protocole ou wrapper.
- **Les nœuds routeurs** : Ils dirigent les données dans le réseau en accord avec le protocole choisi et intègrent la stratégie de routage.
- **Les liens** : Ils connectent les nœuds deux à deux, ce sont eux qui offrent la bande passante. Un lien peut consister en plusieurs canaux virtuels et peut être monodirectionnel ou bidirectionnel.

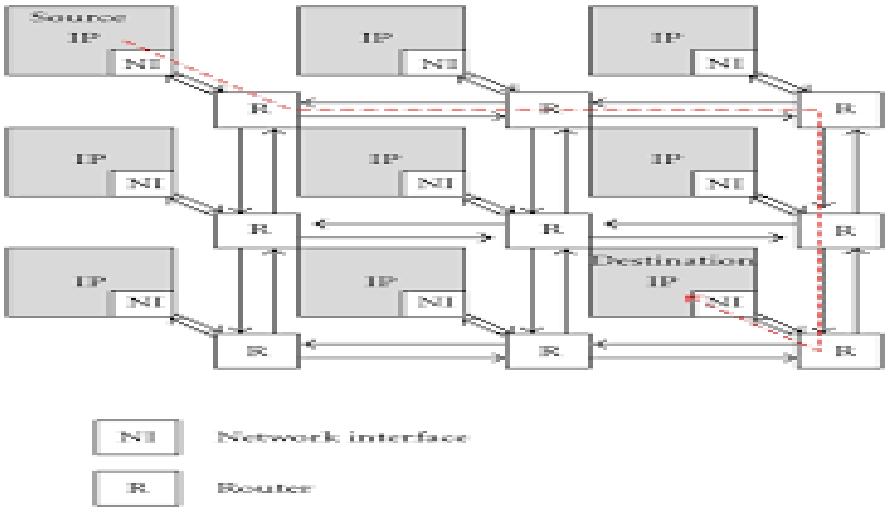


Figure I.11: Topologie de maillage et ses composants [17].

I.4.4. Modes de commutation

La communication dans un réseau basée sur le mécanisme de commutation, ce mécanisme est une manière de transformation des informations sur le réseau.

Deux modes de commutation sont utilisés dans les réseaux sur puce : la commutation de circuit et la commutation de paquets.

- **Commutation de circuit**

Ce mode de commutation consiste à établir un circuit dédié au sein du réseau pour chaque paire émetteur/récepteur [18].

Cela garantit une large bande passante entre les deux ressources et améliore les performances du système lorsque les données échangées sont importantes. Cependant, ce mode de commutation est très désavantageux en termes de ressources et d'énergie, car celles-ci sont expropriées tout au long du processus de transfert. Lorsque la communication se termine, le chemin est dissous [19].

- **Commutation de paquets**

Il existe plusieurs mécanismes de commutation basés sur le principe de la commutation de paquets parmi ces modes on trouve le store-and-forward, le wormhole et le virtual cut-through (VCT).

Le principe de travail dans ces trois modes a divisé le message à transmettre en un ensemble de paquets où chaque paquet lui-même est formé par un ensemble de FLIT.

Chaque flit est stocké dans une file d'attente puis transmis sur le canal approprié. A la réception, le paquet est reconstruit à partir du flit reçu. Ce mode bascule permet un meilleur partage des éléments du réseau car le canal est libéré une fois qu'un flit est envoyé. Cela réduit la latence et améliore les performances du système, mais nécessite un contrôle de flux et de congestion [20,21]. La commutation de circuits est sécurisée, tandis que la commutation de paquets est flexible.

I.4.5. Contrôle de flux

Un protocole de communication définit un ensemble de mécanismes permettant de contrôler la communication entre différentes entités dans un réseau, il permet d'authentifier et de synchroniser les flux de données entre elles, d'éviter de surcharger le réseau et de réguler le trafic.

Les stratégies plus connues sont :

- **Handshake**

La communication est synchrone, où l'expéditeur et le destinataire doivent être synchronisés par le même signal d'horloge, puis l'expéditeur n'envoie plus de paquets et reste en attente jusqu'à ce que le destinataire ne lui ait pas envoyé un ACK indiquant que les anciennes données ont été reçues. Cette technique est donc basée sur le principe de l'attente d'un accusé de réception (ACK). En conséquence, la latence du système augmente et les performances diminuent [22].

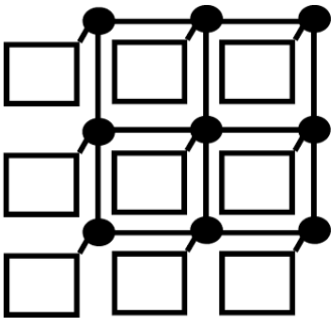
- **Crédit-Based**

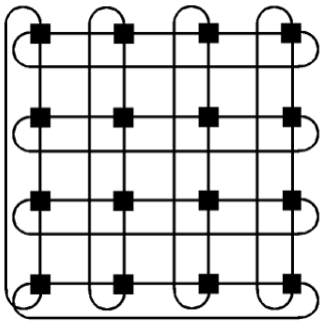
La stratégie consiste à utiliser des crédits, qui indiqueront à l'expéditeur le nombre d'emplacements disponibles dans la mémoire du destinataire. Dans un premier temps, l'émetteur dispose de tous les crédits, puis, lorsqu'il envoie des données, il réduit son nombre de crédits jusqu'à ce que celui-ci devienne nul. Lorsque le récepteur consomme les données qui lui sont envoyées, il notifie à l'émetteur le nombre de positions nouvellement libérées [23].

I.4.6. Topologie

La topologie désigne le graphe des liens entre les différents éléments du réseau. Dans un réseau sur puce, l'un des défis de conception est de choisir la meilleure topologie pour répondre aux exigences de bande passante et de latence de l'application cible avec le coût de puissance et de surface le plus bas, Il existe deux types de topologies régulière et irrégulière, Une topologie se caractérise également par sa forme (linéaire, grille, arbre, tore, multi-grille, hypercube, etc.) et sa régularité ou non.

Dans les réseaux sur puce, Différentes topologies sont proposées dans la littérature pour la conception de NOC, sont décrites ci-dessous.





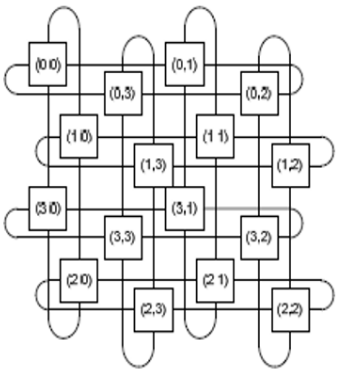


Figure I.12: maillée **Figure I.13:** tore **FigureI.14:** toreplié

I.4.6.1. Topologie maillée

La topologie maillée a été proposée par Kumar [24]. Une topologie maillée se compose de ressources et de routeurs. Un routeur en topologie maillée a des connexions bidirectionnelles vers quatre routeurs voisins et une adresse IP, donc cinq ports d'entrée et cinq ports de sortie. Une topologie maillée peut faciliter le routage.

Le routage XY sous le modèle de trafic de points chauds est un bon routage pour la topologie maillée.

I.4.6.2. Topologie du tore

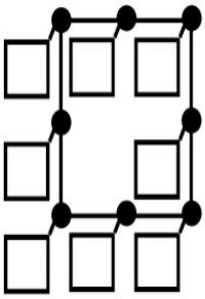
La topologie du tore également appelée k-ary, n-cube est une grille à n dimensions avec k nœuds dans chaque dimension. La topologie en tore a été proposée pour réduire la latence du maillage et garder sa simplicité. Lorsque la latence, la consommation électrique est un critère contraignant, il est préférable d'utiliser la topologie en tore. Bien que l'architecture en tore réduise le diamètre du réseau, les longues connexions enveloppantes peuvent entraîner des retards excessifs.

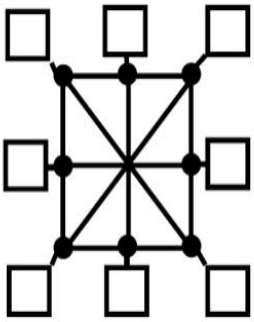
Cependant, ce problème peut être évité en repliant le tore [25]. L'avantage d'utiliser la topologie en tore est une bonne diversité de chemin et un bon équilibrage de charge [26].

I.4.6.3. Topologie du toreplié

Le tore plié a été proposé par Dally [27]. La topologie Folded Torus est similaire à l'architecture maillée, sauf que les fils sont enroulés du composant le plus à droite vers le plus à gauche et du haut vers le bas. La dissipation d'énergie augmente linéairement avec l'augmentation des canaux virtuels. En raison de l'emploi de techniques de commutation de canaux virtuels, la taille de la mémoire tampon est relativement importante.

La principale caractéristique de cette topologie est l'utilisation d'un plus grand nombre de broches pour transmettre des flits de données plus larges généralement sur des canaux filaires. La topologie de tore pliant surmonte la limitation de liaison longue d'un tore 2-D.





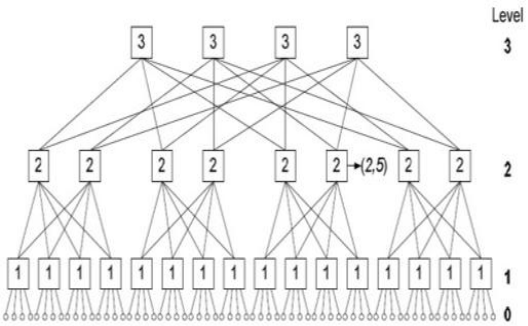


Figure I.15:anneau **Figure I.16:** Spidergon **Figure I.17:**l'arbreàpapillonsgras

I.4.6.4. Topologie en anneau

L'anneau a moins de canaux que les autres architectures et se comporte donc généralement moins bien que les autres topologies NOC. L'architecture en anneau nécessite des canaux virtuels, ce qui double les besoins en mémoire tampon de l'architecture.

I.4.6.5. Topologie Spidergon

L'architecture Spidergon est similaire à un anneau bidirectionnel enrichi par des liens transversaux entre des nœuds opposés. Spidergon évite les impasses en adoptant certaines règles de restriction sur l'utilisation des canaux, d'une manière similaire à l'ordre des dimensions utilisé par l'algorithme de routage Mesh. Spidergon se comporte de la même manière que le maillage même s'il a un plus petit nombre de canaux que le maillage.

I.4.6.6. Topologie de l'arbre à papillons gras

Dans la topologie Butterfly fat-tree [28] , chaque processeur est situé au nœud feuille du fat- tree. Les messages peuvent être échangés entre les éléments de traitement en voyageant de haut en bas sur le réseau fat-tree. Le réseau d'arbres gras peut être considéré comme un réseau d'arbres n-aire étendu avec plusieurs nœuds racines. L'en-tête de paquet tient dans le premier mot, où un octet dans l'en-tête identifie l'adresse de destination et d'autres bits sont utilisés pour le marquage des paquets et les informations de routage.

Les retards du réseau dépendent de la profondeur de l'arbre. Le routage de retournement prend en charge la communication entre les nœuds enfants. L'algorithme de routage de base utilisé dans le NOC basé sur BFT est basé sur une table de routage. La bande passante du Fat-Tree augmente à mesure qu'il se rapproche de la racine.

I.4.7. Les types de réseaux sur puce

Il existe un grand nombre de réseaux sur puce (NOC), Dans ce qui suit nous présenterons quelques exemples de NOC qui nous semblent les plus représentatifs.

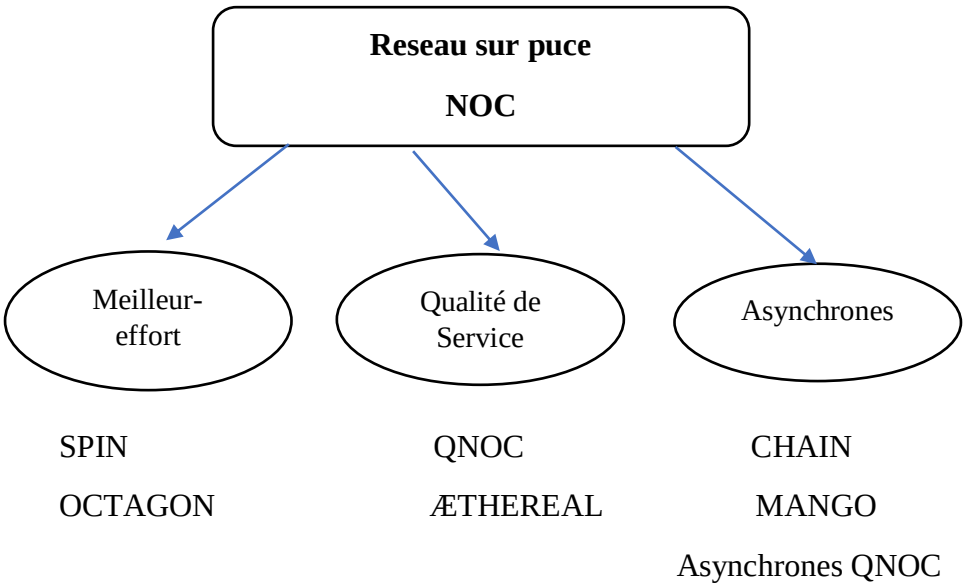


Figure I.18: les types des NOC**.**

II.4.7.1 Meilleur effort NOC

Les réseaux sur puce « meilleur-effort » sont des réseaux conçus dans le but d'avoir des bonnes performances moyennes. Tous les objets communicants dans le réseau se partagent la capacite de trafic du réseau, aucune priorité est accordée.

- **SPIN (Scalable Programmable Integrated Network)**

Le réseau SPIN développé par le laboratoire LIP6 en 2000 [29, 30], ce réseau est un réseau d'interconnexion à commutation de paquets pour systèmes intégrés sur puce.

Le SPIN s'appuie sur une topologie en arbre élargi pour fournir une faible latence grâce à un petit diamètre et un cout de masquage efficace.

Cette technologie fournit un mécanisme de communication très général entre les différents composants virtuels connectés au système. De plus, le nombre de processeurs intégrés dans la bande passante croît linéairement [31] :

Alors que la topologie de l'arbre gras permet à un réseau sur puce d'utiliser efficacement ses ressources physiques, elle n'est pas bien adaptée à l'interconnexion des différents composants d'un système sur puce. Pour répondre à cette restriction, les deux nouvelles versions de SPIN utilisent une topologie de grille et la méthode de routage déterministe distribuée X-first. DSPIN [32], la première nouvelle version de ce réseau synchrone, est

composé de nœuds de routage qui ont chacun leur propre minuterie de synchronisation et sont connectés par des FIFO bi-synchrones.

Une limitation de cette implémentation est l'introduction de latences significatives pour synchroniser les communications à travers les différents domaines de l'horloge des nœuds de routage. La deuxième nouvelle version de SPIN, connue sous le nom d'ASPIN [33], est composée de nœuds de routage asynchrones qui communiquent entre eux en utilisant un protocole de poignée de main insensible au temps. Un avantage de cette implémentation est qu'elle réduit la latence de transmission d'un paquet car seules ses interfaces réseau doivent être synchronisées avec des composants synchronisés.

- **OCTAGON**

Le réseau OCTAGON a été proposé par STMicroelectronics et l'Université de Californie à San Diego en 2001 [34], il est utilisé La topologie anneau octogonal avec des liens supplémentaires entre routeurs diamétralement opposés, La dimension d'une connexion est variable (N bits de données et 3 bits de contrôle).

L'architecture du réseau est conçue pour répondre aux exigences de bande passante. Cette architecture a un faible encombrement pour 8 éléments (la communication entre n'importe quelle paire de routeurs peut être complétée après au moins deux passages), un algorithme de routage simple, et un débit plus efficace qu'une barre transversale.

Selon le choix de l'arbitre de connexion, le mécanisme de commutation peut être circuit ou paquet. Avec la commutation de paquets, la communication entre les ressources s'effectue en routant les paquets à l'aide d'un système de routage distribué et adaptable. Lorsque la commutation de circuit est utilisée, cependant, un arbitre de réseau établit le chemin entier entre les routeurs source et destination pour un nombre spécifique de cycles de temps [35].

II.4.7.2 Qualité de service NOC

Les réseaux sur puce « qualité de service » intègrent des services garantis qui permettent de réserver des ressources de communication pour différents flux afin de garantir leurs débits et/ou leurs latences. Ce type de réseau est très favorable pour les applications temps-réel telles que le traitement de la vidéo et de la voix.

- **QNOC (Quality-of-Service Network-on-Chip)**

Le réseau QNOC est proposée par l'Institut Technologie du Technion [36],La topologie du ce réseau est une maille a deux dimensions, elle peut être irrégulière, Le routage est déterminé par la source et utilise une technique X-Y améliorée pour permettre le routage dans une grille 2D non régulière.

Il a pour caractéristique de séparer les trafics en quatre classes et de les répartir sur des canaux virtuels différents avec un ordre de priorité [37]. Ces classes de trafics sont ordonnées en fonction de leur nature :

- **Signalisation** : les messages urgents et les paquets courts qui sont de type contrôle ou interruption. La plus haute priorité leur est donnée pour leur garantir une courte latence.
- **Temps-réel** : les communications nécessitant une garantie en bande passante et latence, tels que les flux audios ou vidéo. Cette classe de trafic bénéficie d'une grande bande passante.
- **Lecture/Écriture** : les communications de sémantique bus et les accès courts à de la mémoire ou à des registres.
- **Transfert de gros bloc de données** : les transferts du type DMA. Cette classe de trafic a la plus faible priorité.

Enfin, Un ordonnancement préemptif permet aux paquets appartenant à un trafic plus prioritaire d'utiliser les ressources utilisées jusque-là par un trafic moins prioritaire.

- **ÆTHEREAL**

Le réseau ÆTHEREAL est proposé par les laboratoires de recherche de Philips (Eindhoven aux Pays-Bas) [38], ce réseau est base sur une topologie non régulière et utilisant un algorithme de routage par la source.

Les ports réseau d'Ethereal prennent en charge une variété de protocoles de communication conventionnels (OCP, AXI et DTL). La caractéristique principale d'Ethereal est qu'il a des routeurs avec deux niveaux différents de qualité de service. Contrairement au niveau de meilleur effort, qui ne fournit aucune garantie, le niveau de service garanti fournit des garanties sur les dettes et la latence. Les routeurs utilisent l'approche de commutation Wormhole pour fournir des paquets avec un niveau de qualité de service plus élevé. Pour transporter des paquets de type service garanti, une technologie de commutation de circuit basée sur le multiplexage temporel, ou TDM, est utilisée. Cette technique de commutation consiste à créer un circuit logique en réservant des périodes de temps aux routeurs.

Ce mécanisme garantit que les paquets voyageant sur l'un de ces circuits arrivent à destination sans entrer en collision avec d'autres paquets. Le principal inconvénient de cette approche est que la latence garantie est inversement proportionnelle à la dette garantie. En conséquence, ce mécanisme de commutation empêche l'établissement d'une connexion à faible débit et à faible latence. Cette forme de connexion est toutefois vitale, car elle permet le maintien approprié des communications interrompues.

II.4.7.3 Asynchrones NOC

Les réseaux sur puce asynchrones sont des réseaux conçus et implémentes en utilisant les méthodologies de conception asynchrones. Il n'existe donc pas d'horloge globale dans ces réseaux. L'avantage de ce type de réseau est d'éviter des problématiques concernant `a l'horloge dans la conception des Soc complexes.

- **CHAIN (CHip Area INterconnect)**

Le réseau CHAIN est développé par l'Université de Manchester [39]. C'est un réseau implémente entièrement en logique asynchrone de type insensible aux délais [40] et ce réseau possède une architecture entièrement asynchrone [41], Il a utilisé des liens insensibles à la latence ainsi qu'un protocole à double rail en quatre phases. Il utilise la commutation de paquets et le routage des sources. Il est en charge d'un trafic BE. Il peut réduire sa consommation par l'absence de l'horloge.

- **MANGO (Message-passing Asynchronous Network-on-chip providing Guaranteed services over OCP interface)**

Le réseau MANGO est proposé par l'Université Technique du Danemark [42]. Il est un Noc sans horloge [43] et il utilise une topologie en forme de grille et qui possède des interfaces réseau conformes à la norme OCP. Comme Æthereal, les nœuds de routage de MANGO fournissent les deux niveaux de qualité de services meilleur effort et service garanti. Une connexion à service garanti est établie en réservant une suite de canaux virtuels et en appliquant une politique d'arbitrage appelé ALG [43].

Le but de cette politique d'arbitrage est d'accorder l'accès à un lien de communication tout en respectant la latence et les dettes associées aux canaux virtuels connectés à ce lien. Le principal avantage de cette stratégie par rapport au multiplexage temporel d'Ethereal est que les garanties de latence sont combinées avec des garanties de débit. Une lacune de cette stratégie est qu'elle ne permet pas une utilisation complète de la bande passante disponible pour les connexions garanties.

- **Asynchrones QNOC**

Suite à son réseau QNOC synchrone, l'Institut Technologie du Technion a proposé une implémentation asynchrone de QNOC [44]. La topologie du réseau est aussi une maille à deux dimensions et l'algorithme de routage est aussi de type source, déterministe. Le routeur asynchrone a été comparé avec le routeur synchrone avec la même fonctionnalité, et la même bibliothèque de cellules. Donc, la surface de routeur asynchrone est inferieure a la surface du routeur synchrone tandis la performance est supérieure. Par exemple, un routeur asynchrone

avec 4 niveaux de services occupe une surface 0, 47mm2 et il a un d´débit maximal de 75, 2Mflits/s tandis qu'un routeur synchrone avec le même nombre de services occupe 0, 96mm2 et possède un débit maximal de 67, 6Mflits/s.

I.5. Conclusion

Dans ce chapitre nous avons présenté les systèmes sur puce et les différents types de technologie « FPGA et ASIC », on a parlé de leurs architectures et leurs différents types. On a vu aussi la différence entre ces deux technologies. Ensuite nous avons abordé le nouveau paradigme qui est le réseau sur puce. On a vu les trois éléments de base du réseau sur puce et quelques concepts liée à ce réseau.

CHAPITRE II : Les QNOC et les algorithmes de routage dans les Noc.

II.1. Introduction

Le réseau sur puce est une technique de conception du système de communication entre les cœurs sur les Systèmes sur puce. Dans ce réseau, il existe plusieurs types, parmi ces types on trouve le réseau sur puce avec une qualité de service QNOC, qu'il est basé sur des topologies d'arbre gras, maillage régulier et irrégulier.

L'élément de base dans le réseau QNOC est le retour Q-Switch. Ce retour transmis les paquets des ports d'entrée aux ports de sortie.

Pour placer les éléments du réseau dans une interface reconfigurable, le réseau QNOC applique certain nombre de règle.

Pour faire une bonne communication entre les différents cœurs d'un réseau sur puce on utilise des algorithmes de routage qui détermine un chemin emprunté par un paquet entre la source et la destination.

Les algorithmes de routage sont classés sur 4 classes, parmi ceux-ci, nous avons mis évidence le routage XY adaptatif.

II.2. L'architecture de QNOC

II.2.1. La topologie de QNOC

Les réseaux sur puce sont constitués de routeurs interconnectés par des liaisons point à point. La topologie peut varier en fonction des besoins du système et de la taille et de l'emplacement des modules. Des topologies d'arbre gras, de tore plié et de maillage régulier ont été proposées pour NOC (Fig. II.1).

Ils proposent une topologie de maillage irrégulier qui correspond le mieux à la structure plane et rectangulaire généralement irrégulière des plans d'étage Soc courants.

Chaque module du système est connecté à un routeur (Fig. II.1c) via une interface standard, où la bande passante est adaptée aux besoins de communication du module.

Le taux de données de chaque liaison entre routeurs est ajusté de façon similaire pour tenir compte du trafic entrant et répondre aux exigences de qualité de service au niveau de la liaison.

La largeur de bande de l'interface et de la connexion peut être ajustée en changeant soit le nombre de fils, soit la fréquence des données.

Un module peut également être connecté au réseau à l'aide de plusieurs interfaces. Le routage est effectué sur les chemins fixes les plus courts, en utilisant une discipline X-Y dans laquelle chaque paquet est acheminé d'abord dans une direction « X », puis dans la dimension perpendiculaire, ou vice versa [45]. Ce schéma conduit à une solution simple et rentable implémentation de routeur.

Le trafic réseau est ainsi réparti de manière non uniforme sur les liaisons maillées, mais la bande passante de chaque liaison est ajustée à sa charge attendue, ce qui permet d'obtenir un niveau d'utilisation des liaisons à peu près égal sur la puce.

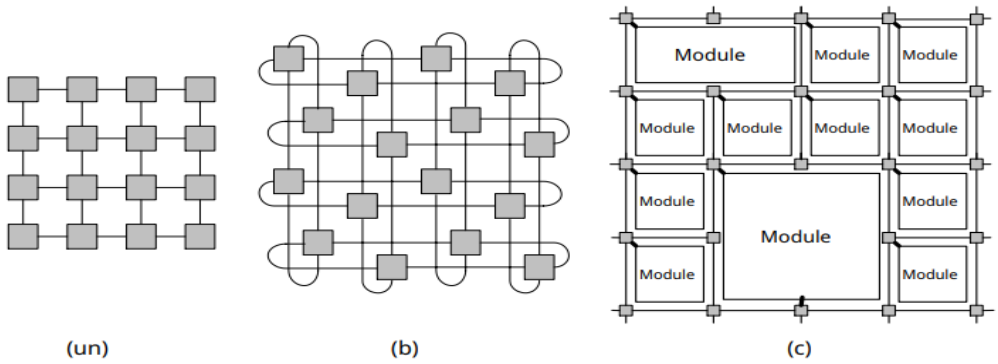


Figure II.1: Topologies Noc : (a) maillage régulier ; (b) tore plié ; (c)topologie personnalisée de maillage irrégulier [47].

II.2.2. Les niveaux de service QNOC

L'objectif principal d'un réseau d'interconnexion sur puce est de répondre à toutes les demandes de communication de modules hétérogènes au sein de la puce. Un QNOC devrait remplacer non seulement les bus partagés, mais également d'autres types de câbles et d'interfaces inter modulaires dédiés. Ils identifient quatre différents types d'exigences de communication et définissons les niveaux de service (SL) appropriés pour les prendre en charge : signalisation, temps réel, lecture/écriture et transfert de blocs.

Deux classes de service ont été précédemment proposé dans ce contexte dans : best-effort et débit garanti [46].

Signalisation couvre les messages urgents et les paquets très courts qui reçoivent la plus haute priorité dans le réseau pour assurer la latence la plus courte. Ce niveau de service convient aux interruptions et aux signaux de commande et évite d'avoir à leur consacrer des câbles spéciaux à usage unique.

Temps réelle niveau de service garantit la bande passante et la latence aux applications en temps réel, telles que le traitement audio et vidéo en streaming. Ce service (comme tous les autres) est basé sur les paquets (et n'emploie pas de circuits virtuels) ; un certain niveau maximal de bande passante peut être alloué à chaque liaison en temps réel et il ne doit pas être dépassé. Ceci est réalisé soit par la conception de chaque module, soit par des circuits d'application dans le réseau.

Lecture/écriture (RD/WR) Le niveau de service fournit une sémantique de bus et est donc conçu pour prendre en charge les accès courts à la mémoire et aux registres.

Bloc-transfert le niveau de service est utilisé pour les transferts de messages longs et de gros blocs de données, tels que le remplissage du cache et les transferts DMA.

Un classement de priorité est établi parmi les quatre niveaux de service, où la signalisation a la priorité la plus élevée et le bloc-transfert la plus basse. Ci-dessous, nous décrivons une planification de communication préemptive dans laquelle les données d'un paquet de priorité plus élevée sont toujours transmises avant celles d'un niveau de service inférieur (un round-robin est utilisé dans chaque niveau de service). Ainsi, les niveaux de service sont simplement mis en œuvre au moyen d'un mécanisme de priorité. Des niveaux de service supplémentaires peuvent être définis si vous le souhaitez, tant qu'un classement de priorité est respecté. Par exemple, le niveau de service RD/WR peut être divisé en sous-niveaux RD/WR normal et urgent [47].

II.2.3. La communication QNOC

Les paquets transportent des informations de routage, de commande et de charge utile. La figure II.18 montre le format de paquet de base. Le champ Adresse de routage cible (TRA) contient l'adresse requise pour le routage. Le champ de commande identifie la charge utile, en spécifiant le type d'opération. Le reste est une charge utile de longueur arbitraire, y compris des informations de contrôle spécifiques à l'opération telles que l'identification de l'expéditeur.

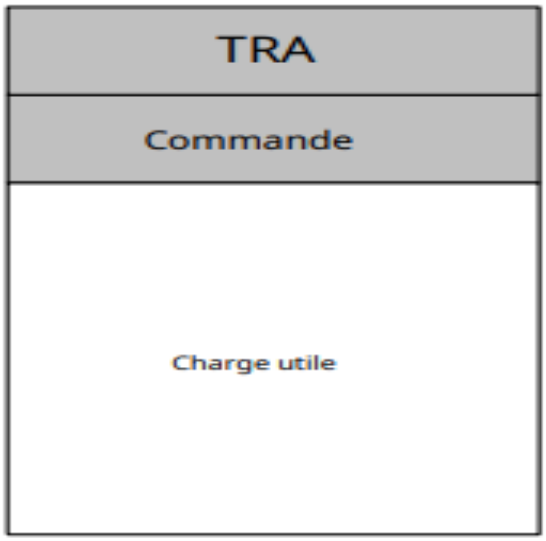


Figure II.2: Format de paquet [47].

Le paquet est divisé en plusieurs flits [48] et transmis par les signaux Data_o (voir tableau 1). Le transfert flit sur le lien est contrôlé par handshake. Les flits sont classés dans les types suivants :

- FP (paquet complet) : un paquet d'un seul flit.
- EP (fin de paquet) : dernier flit d'un paquet.
- BDY (corps) : un non-dernier volte-face.

Ainsi, tous les flits sauf le dernier dans un paquet sont étiquetés BDY. Le premier flit d'un paquet peut être détecté comme le premier flit valide suivant un flit FP ou EP (cette identification déclenche le mécanisme de routage).

II.2.4. Routeurs QNOC

Les routeurs se connectent à cinq liens maximum (figure **II.3**), conçus pour une interconnexion planaire à quatre mailles voisines et à un module de puce. Le routeur transmet les paquets des ports d'entrée aux ports de sortie. Les données sont reçues par flit. Chaque flit arrivant est d'abord stocké dans un tampon d'entrée. Au premier flux d'un paquet, le routeur détermine à quel port de sortie ce paquet est destiné. Le routeur programme alors la transmission pour chaque flit sur le port de sortie approprié [48]. **La figure II.4** illustre le flit de données dans un routeur à quatre liaisons.

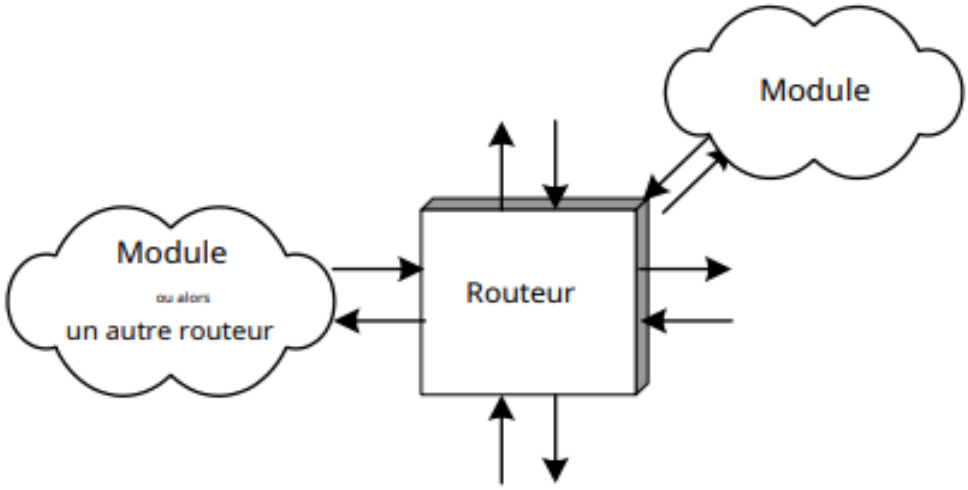


Figure II.3: Le routeur a jusqu'à cinq liaisons et peut se connecter à des routeurs maillés voisins ou à des modules à puce [47].

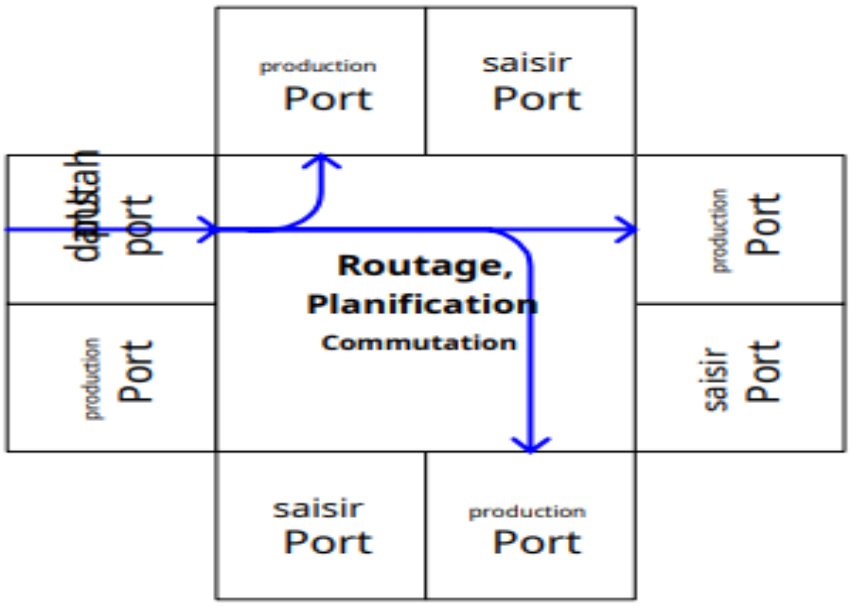


Figure II.4: Routeur—flux de données [47].

Il existe des tampons distincts pour chacun des quatre niveaux de service ("direct buffer mapping"). Des tampons relativement petits sont alloués à chaque niveau de service, capables de stocker seulement quelques flux (il s'agit d'un paramètre de conception réglable). L'algorithme de routage est invoqué lorsque le premier flit d'un paquet est reçu. L'algorithme utilise une fonction de routage simple. Par exemple, le routage relatif est utilisé pour X–Y routage. Les informations de routage pour chaque niveau de service et chaque port d'entrée sont stockés dans la table de routage actuelle (**Figure II.5**), jusqu'à ce que le flit de queue du paquet soit reçu, traité et livré. Lorsqu'un flit est transmis d'une entrée à un port de sortie, un tampon devient disponible et un crédit tampon est envoyé retour au routeur précédent sur des fils hors bande séparés. Chaque port de sortie d'un routeur est relié à un port d'entrée d'un routeur suivant via une liaison de communication. Le port de sortie maintient le nombre de créneaux flit disponibles pour chaque niveau de service dans la mémoire tampon du port d'entrée suivant. Ces numéros sont stockés dans le prochain état du tampon (NBS). Le nombre est décrémenté lors de la transmission d'un flit et incrémenté lors de la réception d'un crédit tampon du routeur suivant. Lorsqu'une place est disponible, le port de sortie planifie la transmission des flit qui sont mis en mémoire tampon aux ports d'entrée et attendent la transmission via ce port de sortie [49].

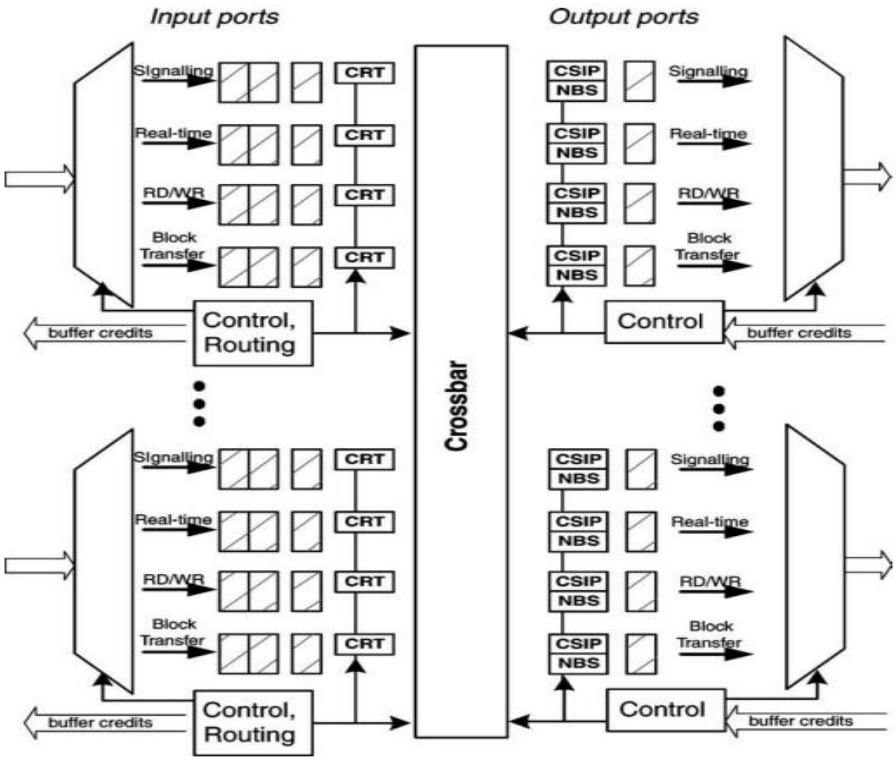


Figure II.5: Architecture du routeur [47].

Dans le réseau QNOC l'élément de base est le routeur Q-switch, Son architecture est présente ci-dessous.

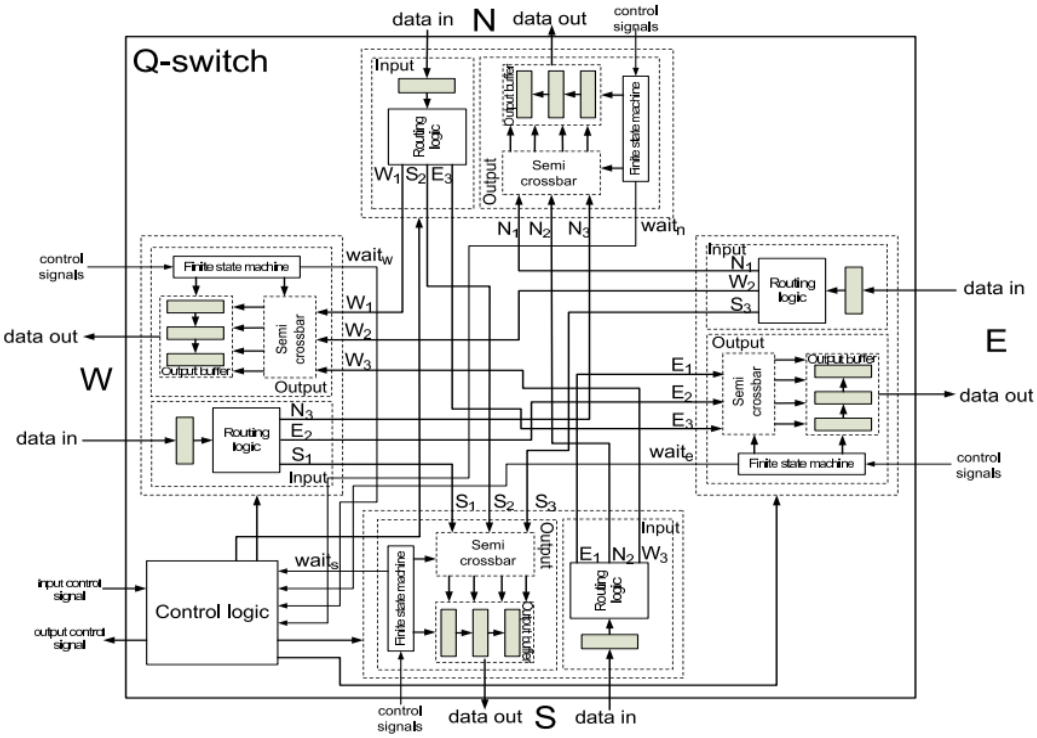


Figure II.6: Architecture du routeur Q-switch d'un réseau QNOC [49].

Tous les paquets de données arrivant à un Q-switch sont initialement chargés dans ces registres d'entrée. Une fois que les paquets ont été reçus et stockés dans ces registres d'entrée, les blocs de calcul "Logique de routage" déterminent leurs directions ultérieures. Les paquets sont envoyés aux blocs "Logique de sortie" sur la base des directions calculées par les blocs "Logique de routage". Ces blocs de départ déterminent l'ordre de départ pour tous les paquets arrivant au même port de départ.

Lorsque trois paquets arrivent en même temps de trois directions différentes pour être achemine à la même destination, ces trois paquets doivent partager la même destination. Le rôle du bloc "Logique de sortie" est de s'assurer que tous les paquets qui ont été envoyés à destination sont achemine en toute sécurité. Ce mécanisme d'arbitrage est utilisé lorsque trois colis arrivent en même temps dans un port de départ (**figure II.7**). Par exemple, un port à l'Est ne peut accepter que les colis arrivant des directions Nord, Ouest et Sud.

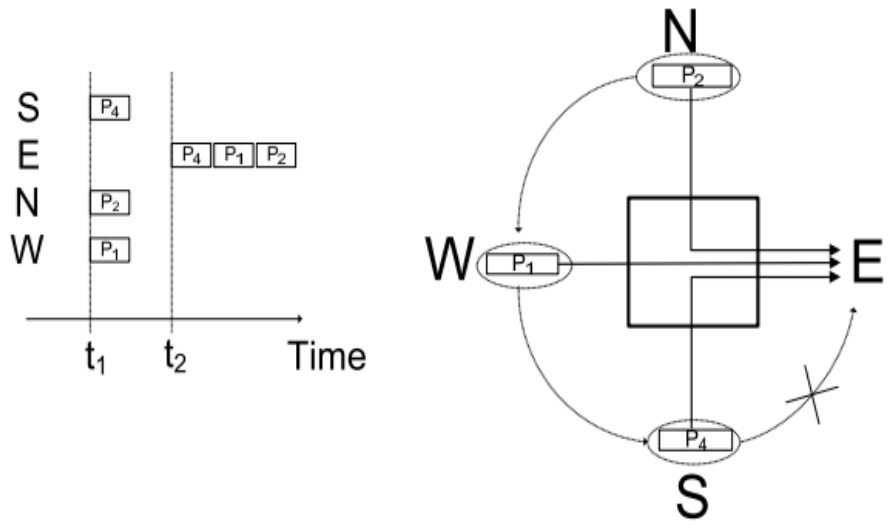


Figure II.7: Politique d'arbitrage basée sur la règle de priorité à droite [49].

Lorsque la règle de priorité à droite est appliquée à ces entrées, les paquets qui arrivent du sud ont la priorité la plus élevée, suivis par ceux qui arrivent de l'ouest, et enfin ceux qui arrivent du nord. Le premier scénario correspond à un Q-switch qui reçoit actuellement quatre paquets de quatre directions différentes, chacune avec une adresse de destination différente. Tous les paquets sont immédiatement transférés vers des routeurs proches en raison de la latence introduite par le routeur, ce qui équivaut à deux cycles de temps. La deuxième situation illustre la réception de 4 paquets parmi lesquels 3 possédants la même adresse de destination. Ces paquets sont reçus à un instant et sont transmis à partir de l'instant.

Ces deux paquets, quittent donc le routeur dans les 2 cycles d'horloge suivant. En effet, la principale raison pour laquelle un réseau QNOC ne repose pas sur la technique d'aiguillage wormhole routing est dû au fait que la structure des paquets de données à transmettre avec cette technique est composée de flux.

II.2.5. Interface QNOC

L'interface réseau connecte les modules au réseau. Il mappe une variété de transactions sur les quatre niveaux de service QNoC structurés. Par exemple, il masque le caractère de commutation de paquets du réseau lorsqu'un module doit accéder à un autre en utilisant le bus et la sémantique de lecture/écriture conventionnelle.

II.3. La performance de QNOC

Une mesure de performance est une quantité mesurable qui saisit exactement ce que l'on veut mesurer. Elle peut prendre plusieurs formes. Il n'existe pas de définition universelle de la mesure de performance parce qu'elle dépend du système, ce qui nécessite une compréhension

approfondie de ce dernier et de ses utilisateurs, ainsi qu'une compréhension des contextes expérimentaux [50].

Les Mesures de performances Réseau sont basé sur :

- ✓ Le débit de donnée.
- ✓ La latence.

II.3.1. Le débit de donnée

Le débit ou la bande passante maximale (throughput) d'un routeur Q-switch au format de données n bit, fonctionnant à la fréquence f et permettant à un nombre maximum de modules de calcul de communiquer simultanément (4 modules situés à chaque adresse d'un routeur Q-switch) est définie par les expressions suivantes [49] :

$$\textbf{Throughputmax[Gbps]} = 4 * n * f$$

$$\textbf{Throughputmax [paquet / clk]} = 4$$

II.3.2. La latence

La latence est initialement déterminée à l'aide de critères statistiques. Il se réfère principalement à la position de l'émetteur par rapport au récepteur du réseau. Plus le nombre de liens et de nœuds à explorer est élevé, plus la latence est grande, principalement en raison du temps passé à naviguer dans les nœuds. La latence peut également changer de manière dynamique. Cela est dû au fait que le réseau est partagé par plusieurs paquets en même temps.

II.4. Les Conditions de placement dans un réseau QNOC

Dans la construction d'un réseau QNOC, les éléments du réseau sont instanciés dans une surface reconfigurable. Dans une première étape, les routeurs du réseau QNOC sont placés dans la zone reconfigurable. Ensuite, des modules de calcul sont localement placés au sein du réseau par substitution de routeurs initialement placés. Si le placement d'un module nécessite la substitution de plusieurs routeurs du réseau, alors le module nouvellement placé hérite de toutes les adresses des routeurs substitués [51].

Le réseau QNOC respecte un certain nombre de règles de placement [52] :

- **Règle No1 :** Chaque module (élément de calcul - PE) doit avoir au moins un accès au réseau via un des ports parmi ses routeurs avoisinants ou l'entourant.
- **Règle No2 :** Tous les modules PE communiquent à travers les routeurs CU.
- **Règle No3 :** Les routeurs Unités de communication sont connectées, soit à d'autres routeurs, soit à des PE. Chaque CU est connecté au moins à un routeur de type différent à lui-même (CU classique à un CU « qui-cède-le-passage » et vice versa).

- **Règle No4 :** Chaque module PE est entouré au maximum dans ses quatre directions part trois modules PE.
- **Règle No5 :** Tous les modules, placés dynamiquement ou statiquement, étant entourés dans leurs quatre directions uniquement par les routeurs CU sont toujours joignables (règle DyNoC).
- **Règle No6 :** Entre tous les modules PE, il doit exister au moins un chemin leur permettant des échanges de paquets de données.

Avant le placement dynamique d'un module de calcul, le routeur ou les routeurs Q-switch de substitution vident leurs tampons et convertissent en mode "module" pour arrêter d'accepter les paquets de données des routeurs voisins. D'autre part, pour le routage de leurs paquets de données ultérieurs, ces derniers prennent en compte le nouveau statut du routeur ou des routeurs en question, traitant la zone de routeur remplacée comme un obstacle réseau à délimiter.

II.5. Les algorithmes de routage

L'algorithme de routage détermine le chemin emprunté par un paquet entre la source et la destination. Dans une architecture de communication, le routage joue un rôle très important ou le meilleur algorithme de routage donne la meilleure performance.

L'algorithme de routage est nécessaire de faire des compromis entre une utilisation optimale des liens de communication et un algorithme simple qui peut être facilement mis en œuvre sur silicium.

On a plusieurs algorithmes de routage sont classes comme le montre le schéma ci-dessous.

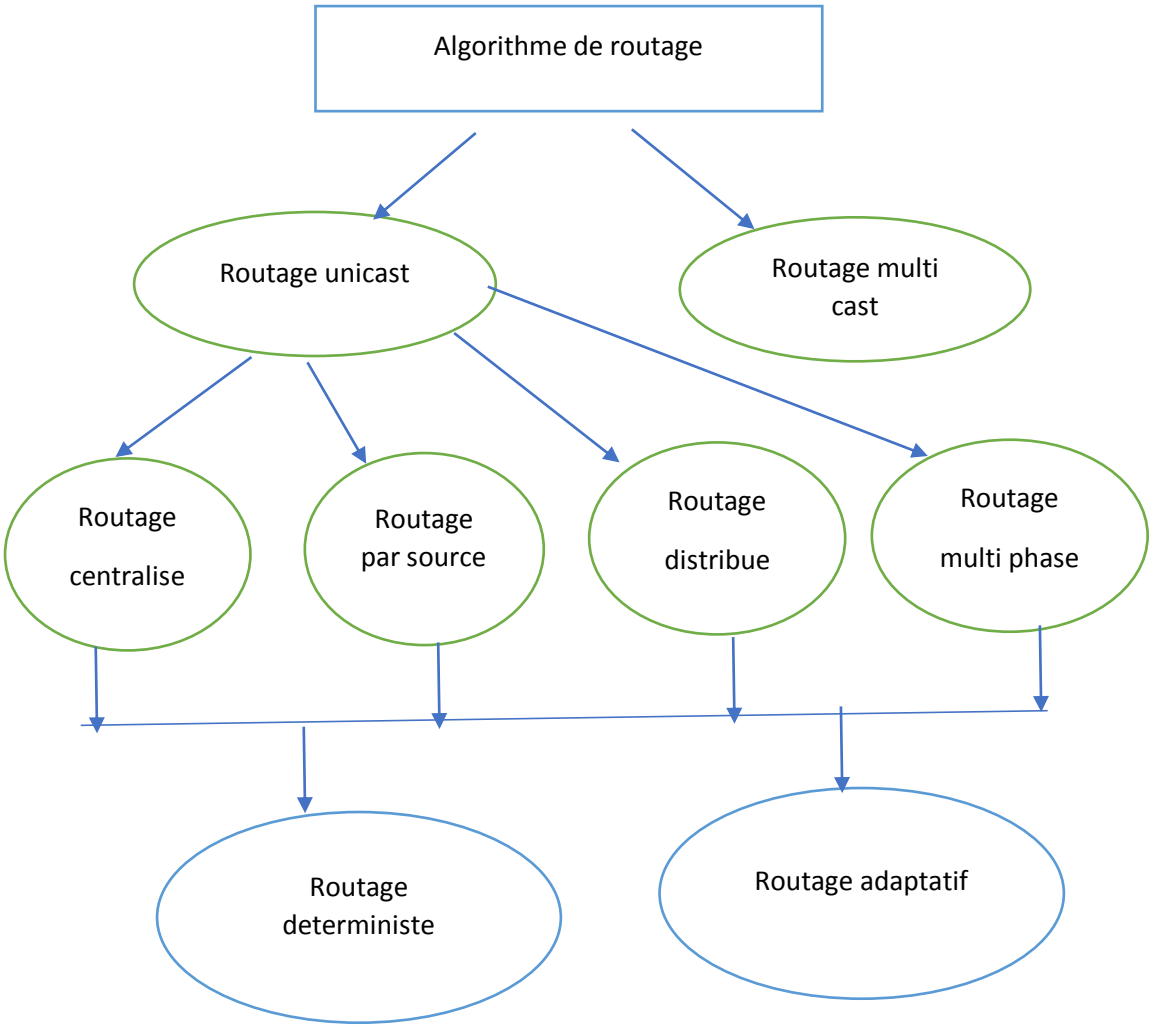


Figure II.8: les algorithmes de routage.

II.5.1. Routage unicast ou multicast

Selon le nombre de destination, on trouve deux classes le routage unicast et le routage multicast. Dans le routage unicast les paquets d'un même message ont une destination unique (ils sont envoyés d'un seul nœud de source vers un seul nœud de destination).

Et dans le cas de routage multicast les paquets ont des destinations multiples (ils sont envoyés d'un seul nœud de source vers plusieurs nœuds de destination) [54].

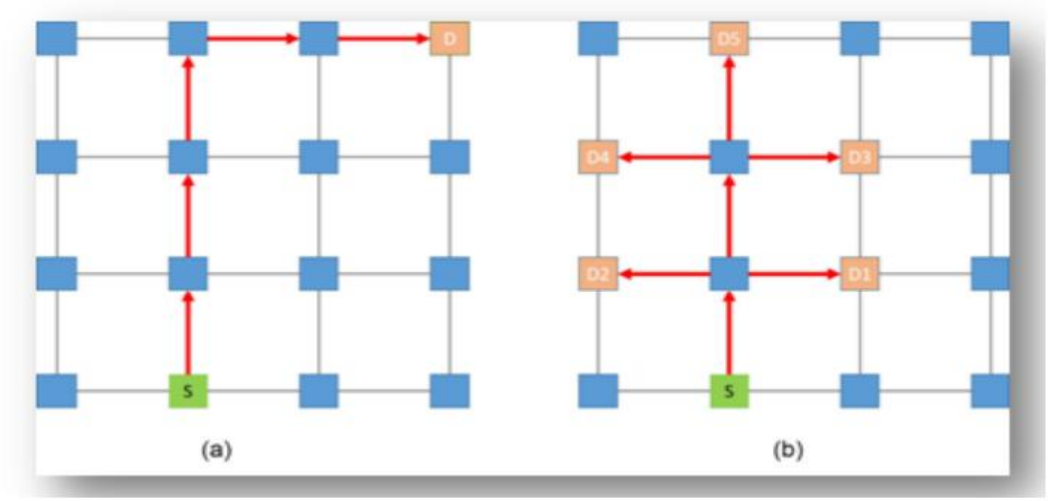


Figure II.9: Routage unicast vs multicast [54]**.**

II.5.2. Routage centralise ou source ou distribué ou multi phase

Selon les décisions de routage les algorithmes de routage unicast divisés en quatre catégories : le routage centralisé, le routage par la source, le routage distribué et le routage multi-phase.

Dans un routage centralisé, c'est un nœud qui possède toutes les informations sur l'état du réseau. Ce nœud est donc en mesure de calculer à chaque instant le chemin optimal entre deux nouds. Ainsi tout nœud source désirant établir une connexion doit s'adresser au nœud "principal", e qui augmente le temps pour calculer une route. De plus, il existe un problème de fiabilité important. En effet, si ce nœud de routage venait à être hors service, ou si un des liens le reliant au reste du réseau était coupé, il y aurait alors un impact sur le bon fonctionnement du réseau [55].

Dans le routage source, toutes les décisions de routage sont prises à l'intérieur du nœud source avant l'injection de tout paquet dans le réseau. A cet effet, chaque source contient des listes ou des tableaux qui contiennent des informations sur l'itinéraire complet pour atteindre toutes les autres ressources dans le réseau.

Afin d'acheminer un paquet à travers le réseau en utilisant le routage source, la ressource expéditrice consulte sa table de routage pour obtenir le chemin complet pour l'accès à la destination souhaitée. Ce chemin est alors écrit dans le champ dédié dans l'entête du paquet. Le paquet doit suivre le chemin en traversant le réseau vers a destination. Chaque routeur qui reçoit ce paquet lit le champ de chemin dans l'entête du paquet et le transmet au port de sortie destiné. Contrairement dans le routage distribué, un paquet ne contient que l'adresse de sa

destination. Un nœud de routage doit don calculer pour chaque paquet vers quel lien de communication il doit l'aiguiller.

La combinaison des algorithmes de routage par la source et la destination nous donne un nouveau routage hybride appelé routage multi-phase [55].

II.5.3. Routage déterministe ou adaptatif

Selon la façon dont un chemin est défini pour transmettre des paquets, on trouve un algorithme de routage soit déterministe ou adaptatif.

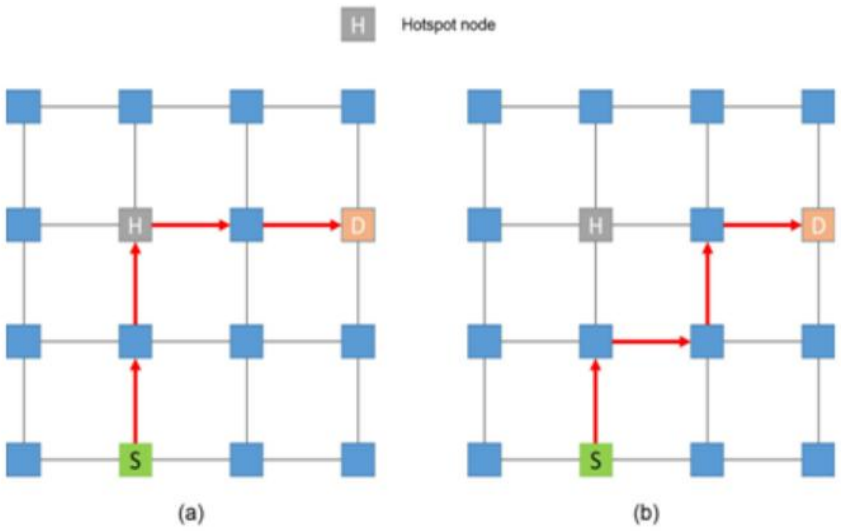


Figure II.10: Routage déterministe vs adaptatif [54].

II.5.3.1. Routage déterministe (statique)

Dans cet algorithme, le chemin des paquets est déterminé à partir des adresses de sa source et de sa destination. Ainsi, le routage déterministe produit le même chemin entre un pair source/destination donnée.

Le routage déterministe est plus simple à mettre en œuvre en termes de logique et de l'interaction entre les routeurs [56]. Il est plus approprié lorsque les changements de données sont très prévisibles et cohérents. Par conséquent, ce routage est préférable pour les réseaux utilisés dans les systèmes spécifiques aux applications (ASIC).

L'avantage est qu'il est simple et peu coûteux à mettre en œuvre. Un autre avantage est sa capacité à éviter le problème de l'inter-blocage.

II.5.3.2. Routage adaptatif (dynamique)

Dans le routage adaptatif chemin de routage est décidé dans les routeurs avant chaque saut des paquets et implique des mécanismes dynamiques d'arbitrage [57]. Donc le routage adaptatif est l'algorithme de routage le plus sophistiqué, dans lequel le chemin qu'un message doit prendre dépend de la situation du trafic et l'état de réseau.

Il existe deux types de routage adaptatif : le routage adaptatif minimal et le routage adaptatif non-minimal. Le chemin emprunté par les paquets est toujours le plus court dans la première catégorie. Lorsqu'il y a plusieurs chemins courts entre chaque paire de sources et de destinations, l'algorithme fonctionne bien. L'algorithme dans ce cas utilise le chemin le moins encombré chaque jour. La deuxième catégorie, d'autre part, prend toujours un chemin non obstrué. La longueur du chemin entre l'émetteur et le récepteur est ignorée par cet algorithme. Typiquement, un algorithme de routage adaptatif choisit toujours l'état du réseau c.à.d. les chemins libres de la congestion, le chemin le plus court est le meilleur [55].

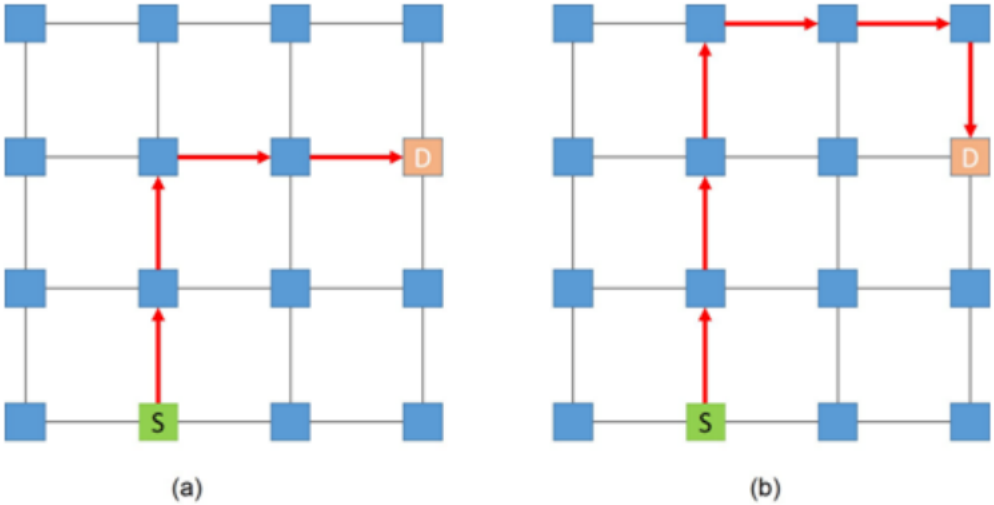


Figure II.11: Routage minimal vs non minimal [54].

L'avantage de routage adaptatif est offre une meilleure tolérance aux pannes, car il permet de calculer dynamiquement le chemin emprunté par message.et aussi il adaptatifs s'adaptent facilement aux différentes topologies.

Cependant, les routages adaptatifs sont plus complexes dans leur mise en place et peuvent facilement gérer des inter-blocages entre les paquets, étant donné que ces derniers peuvent se déplacer librement dans le réseau [58].

II.5.3.3. Les techniques de routage déterministe et adaptatif

Le routage est de type source lorsque c'est l'émetteur qui définit le chemin de routage. Les informations du chemin de routage sont calculées et enregistrées dans une table de routage de l'émetteur. Lorsqu'une source transmet un paquet, elle regarde les informations de routage correspondantes à la destination du paquet, puis elle inclut ces informations dans le flux d'en-tête du paquet. Au contraire, si chaque routeur choisit la destination du paquet en fonction de son adresse cible (et d'autres critères éventuels), il s'agit d'un routage distribue. La décision

de routage est mise en œuvre dans chaque routeur soit en consultant les adresses cibles dans une table de routage ou en exécutant une fonction de routage matérielle [59]. Avec cette méthode, chaque routeur contient une table de routage pré-définie ou des logiques de routage dont l'entrée est l'adresse de destination du paquet et la sortie est la décision de routage. Lorsque le paquet arrive au port d'entrée du routeur, le port de sortie est cherché dans la table ou calcule par la logique de routage en fonction de l'adresse de destination.

II.5.3.4. Les algorithmes de routage statique et dynamique

Nous commençons d'abord avec le routage déterministe (statique), Parmi les exemples des algorithmes de ce type nous trouvons les algorithmes de routage XY, ouest en premier (West first) et négatif en premier (négative first).

- **Algorithme de routage ouest en premier (West first)**

C'est un algorithme de routage semi-déterministe. Le paquet est d'abord transmis dans la direction ouest (si cette direction est nécessaire), puis de manière adaptative dans les directions est, nord ou sud [60].

- **Algorithme de routage négatif en premier (négative first)**

Le paquet se déplace d'abord dans les directions négatives (sud et ouest), puis dans les directions positives (nord et est). Il ne permet pas aux paquets d'aller du sens positif au sens négatif [60].

- **Algorithme de routage XY basique**

Cet algorithme parmi les algorithmes les plus connus et les plus utilisés pour le routage déterministe sur les topologies en grille 2D car il est simple à réaliser.

Dans le routage XY basique ou statique, le paquet est d'abord acheminé sur la dimension X puis sur la dimension Y, ainsi le paquet ne peut effectuer qu'un seul changement de direction. La décision du routage d'un paquet est prise en fonction de l'entête du paquet qui contient, entre autres, l'adresse du nœud destinataire. Cette technique de routage est simple à mettre en œuvre, permettant ainsi une intégration compacte et rapide. Ce routage exclue le risque de deadlock [60,61].

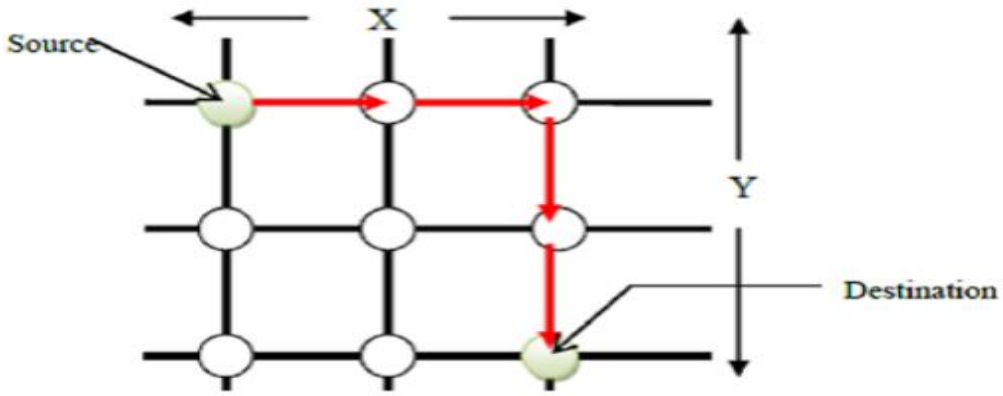


Figure II.12: Routage XY [55].

Apres voir pris des exemples des algorithmes de routage statique (déterministe), nous mettrons en évidence les algorithmes de routage dynamique (adaptatif).

Parmi les exemples d'algorithmes de routage dynamiques nous pouvons citer les algorithmes fully-adaptive, odd-even, congestion look-ahead, routage xy adaptative.

Parmi ceux-ci, nous avons mis évidence le routage XY adaptatif car le réseau QNOC utilise un algorithme de routage basé sur un algorithme XY modifié ou bien adaptatif.

L'algorithme XY ne peut pas être utilisé pour le routage des paquets dans sa version initiale pour un réseau reconfigurable, car il n'est pas adapté à des structures irrégulières du réseau qui peuvent se produire lors d'un placement dynamique des modules de calcul au sein du réseau.

En effet, pour certaine structure irrégulière des routeurs d'un réseau reconfigurable, le routage de paquets de données basé sur l'algorithme XY est impossible.

Pour éviter ce problème, ils ont développé l'algorithme XY classique et il s'est appelé **routage XY adaptatif**.

- **Algorithme de routage XY adaptatif**

Un routage adaptatif XY est utilisé pour rediriger le flux vers le routeur suivant. Il s'agit d'une version modifiée du routage XY couramment utilisé dans les NOC. Dans une route XY, le routeur récupère le paquet d'abord dans la direction X (horizontale), puis dans la direction Y (verticale). Si le routeur adjacent dans la direction X n'a pas de VC libre pour accepter un paquet, le paquet reste bloqué dans le routeur courant jusqu'à ce qu'un VC du routeur adjacent dans la direction X devienne disponible. Si un paquet ne peut pas être livré dans la direction X, il est acheminé dans la direction Y si le routeur est adjacent.

Dans la direction Y dispose d'un VC libre. Ceci a permis d'améliorer de manière sensible la latence du réseau.

On peut aussi appeler le routage XY adaptatif routage XY dynamique (DYXY),La fonction XY Dynamique (DYXY) est une fonction adaptative. Lorsque le chemin le plus court entre un routeur source et un routeur destination n'est pas unique, la fonction DYXY aide le paquet à choisir son chemin selon la charge en cours du réseau. La fonction XYX est une variété avec de la tolérance aux fautes. En effet, les paquets sont envoyés avec une redondance. Un marquage est introduit dans l'entête des paquets afin de distinguer l'original de celui de redondance. Le paquet original est routé en XY tandis que le paquet de redondance est routé en YX. Une autre variété de la fonction de routage XY est la fonction XY adaptative. Elle détermine le chemin des paquets en fonction de la charge en cours du réseau. Le chemin emprunté n'est pas forcément le plus court [52].

II.6. Conclusion

Dans ce chapitre du réseau sur puce avec qualité de service, on a abordé les différents types du NOC. On a exploré en détail le réseau QNOC, ensuite on a présenté l'architecture de QNOC avec les différentes topologies, la communication des paquets, l'interface de QNOC et le retour Q-Switch. On a vu aussi la performance de ce réseau et les conditions de placement dans QNOC. Après cela, on a cité les différents algorithmes de routage, on a basé sur l'algorithme XY adaptatif, à la fin on a expliqué le processus de conception QNOC.

Chapitre III : conception

III.1. Introduction

Un réseau QNOC est composé d'un ensemble des routeurs, qui sont liés entre eux par des liens deux a deux. La communication dans ce type de réseau est faite par la commutation de paquet et utilise le handshake. L'élément de base dans cette topologie « maillé 2D » est le routeur, dans notre cas est appelé Q-Switch.

Q-Switch utilise l'algorithme XY adaptatif pour acheminer les paquets, Apres plusieurs tests de simulation on a constaté que cet algorithme ne marche pas dans les coins**,** notre proposition est d'amélioré cet algorithme de façon que ce problème soit réglé.

III.2. Les caractéristiques d'un Q-Switch

III.2.1. Rôle de Q-Switch

Le rôle est de passer les messages entre un maximum de 4 éléments de traitements (couples routeurs-IP).

III.2.2. Les composants de Q-Switch

Il existe 3 composants importants qui sont présenté ci-dessous :

- **Registre d'entrée :** Chaque paquet entrant est stocké dans un registre d'entrée. Un composant spécifique appelé logique de routage calcule la direction suivante du paquet. La gestion autorise jusqu'à trois paquets. Les paquets sont transmis à la logique de sortie. Politique d'arbitrage disponible définit la priorité parmi les paquets stockés dans les registres d'entrée du routeur, selon direction du prochain paquet. Cette politique est basée sur des règles de priorité.
- **Logique de sortie :** Le bloc "logique de sortie" du port du routeur Q-switch se compose d'un semi-crossbar, d'un tampon de sortie et d'une machine d'état. Le rôle du semi-crossbar est de commuter 3 entrées vers 4 sorties dans les trois autres directions. Les entrées sont classées par priorité. L'entrée numéro 1 a la priorité la plus élevée, l'entrée 2 a une priorité inférieure et l'entrée 3 a une priorité inférieure. La première sortie du crossbar est la sortie qui se connecte directement au routeur si le routeur voisin n'est pas occupé. La deuxième sortie du crossbar est reliée au premier registre du tampon de sortie, la troisième sortie est reliée au deuxième registre du tampon, et la quatrième sortie est reliée au dernier registre du tampon de sortie [62].

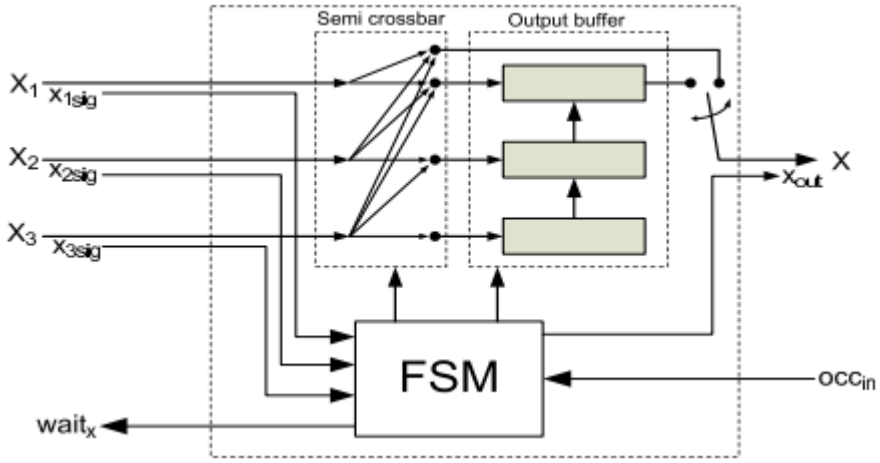


Figure III.1 : l'architecture d'un bloc de sortie [63].

Le tampon de sortie est constitué de trois registres de même taille et égale à la taille du paquet. Dans les situations où plusieurs paquets arrivent à l'entrée du bloc "logique de sortie", le paquet de priorité la plus élevée est transmis directement à la sortie du routeur si un routeur voisin est disponible au moment de la transmission. D'autres paquets de faible priorité sont temporairement stockés dans le tampon de sortie, envoyés un par un après l'envoi du premier paquet.

- **Logique de contrôle** : La sortie Q-switch est commandée par la logique de contrôle sortie. Ces blocs logiques de contrôle sont suffisants pour organiser l'acheminement des paquets, mais uniquement dans les situations où les routeurs voisins ne sont pas occupés. Sinon, des blocs de contrôle supplémentaires sont nécessaires pour gérer de telles situations. Dans une telle situation, il n'est pas possible de transmettre plusieurs paquets en un cycle d'horloge. La logique de contrôle centrale traite ces signaux d'occupation émis. En fait, il envoie un signal de sortie occupé au routeur voisin où le paquet arrive à "1", notifiant que le Q-switch en question est temporairement indisponible et qu'il n'y a pas d'autres paquets en provenance de celui-ci, dans ces directions. Accepté. Ces paquets sont stockés soit dans le registre d'entrée, soit dans le tampon de sortie. Dans l'un ou l'autre cas, la logique de contrôle centrale règle la sortie du signal occupé vers le routeur voisin correspondant sur « 1 » pour indiquer qu'il est temporairement indisponible.

III.3. Etude RKT-Switch

III.3.1. Le Commutateur RKT

Est un périphérique à quatre ports. Ici, les données seraient prises via le module de bouclage. Un flux de données de 24 bits est décodé en données originales de 18 bits.

Le Commutateur RKT est basé sur stocker et retransmettre techniques de commutation. Cette technique est adaptée aux reconfigurables dynamiquement NOC. En effet, dans la réflexion NOC, la PE ou IP peut être mis en œuvre en remplaçant un ou plusieurs routeurs.

III.3.2. L'architecture de Commutateur RKT

Le commutateur RKT se caractérise par :

- Son architecture à quatre directions (nord, sud, est et ouest) adaptée à un NOC maillé 2D.
- Utilise un mode de commutation par paquet.
- Utilise le contrôle de flux Handshak.
- Utilise l'algorithme de routage XY adaptatif.

Les IP et les PE peuvent être directement connectés à n'importe quel côté d'un routeur. Par conséquent, il n'y a pas de port de connexion spécifié pour une IP ou un PE. Les mécanismes de détection proposés peuvent également être appliqués aux NOC utilisant des routeurs à cinq ports avec un port local dédié à une adresse IP.

Chaque direction de port est composée de deux bus de données unidirectionnels (ports d'entrée et de sortie). Chaque port d'entrée est associé à une FIFO (tampons) et à un bloc logique de routage.

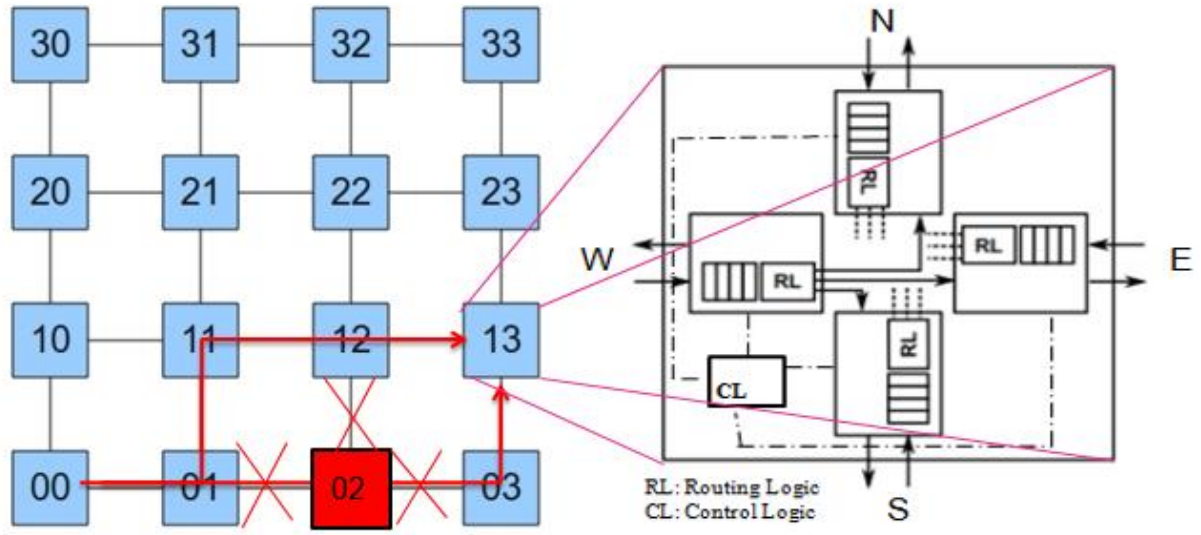


Figure III.2 : architecture du Q-Switch.

III.4. Problématique

III.4.1. Processus de routage

L'algorithme de routage X-Y signifie que les paquets de données sont dirigés vers le destinateur à travers des routeurs selon l'axe X puis selon son axe Y avec la condition que le paquet ne prend pas la direction d'arrivée (On a déjà détaillé cet algorithme dans le chapitre 2).

Quand plusieurs paquets arrivent au même temps à une direction de sortie, on prend jusqu'à 3 paquets au maximum, ces paquets sont sorties par une politique d'arbitrage qui est fondé sur les règles des priorités à droite.

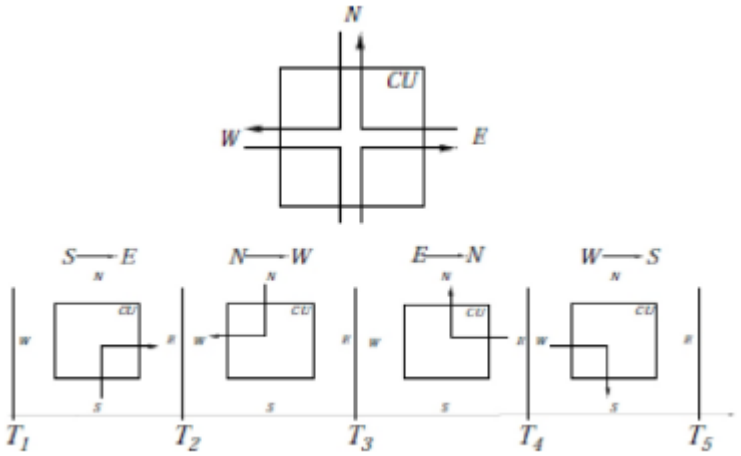


Figure III.3 : Règles des priorités à droite [63].

III.4.2. La zone active

La zone désactivée d'un réseau est le reste du réseau n'appartenant pas à la zone activée.

Si le réseau n'a pas de zones activées (pas de nœuds ou de zones défaillantes), il est entièrement désactivé. Tous les nœuds appartenant à la zone désactivée sont désactivés. Si le réseau a formé une zone active autour du nœud défectueux, avec seulement des nœuds de routage entourant le nœud défectueux.

Les nœuds de routage défaillants changent d'état et deviennent actifs. Nœud appartenant au reste le réseau ne change pas de mode et reste désactivé.

Les nœuds désactivés transmettent les paquets selon l'algorithme XY. D'abord il route les paquets le long de l'axe X, puis le long de l'axe Y, jusqu'à ce que le paquet soit livré à destination. Si le paquet atteint la région active avant d'atteindre sa destination finale, alors

Appliquez ensuite les nouvelles règles de routage.

Les nouvelles règles sont [64] :

R1 : Un nœud pair et activé ne peut pas acheminer des paquets venant de la direction Nord (North) vers la direction Est (East) et vice versa.

R2 : Un nœud impair et activé ne peut pas acheminer des paquets venant de la direction Sud (South) vers la direction Ouest (West) et vice versa.

R3 : Tous les nœuds activés, par défaut, ne peuvent pas acheminer des paquets venant respectivement des directions Nord et Est vers les directions Sud et Ouest.

R4 : Tous les nœuds activés ne peuvent pas par défaut acheminer des paquets venant respectivement des directions Sud et Ouest vers les directions Nord et Est.

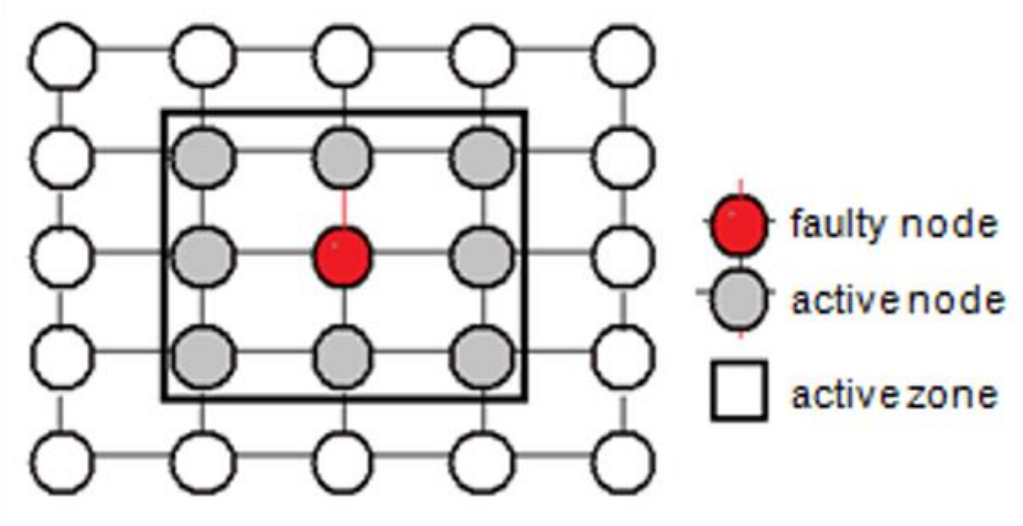


Figure III.4 : une zone active.

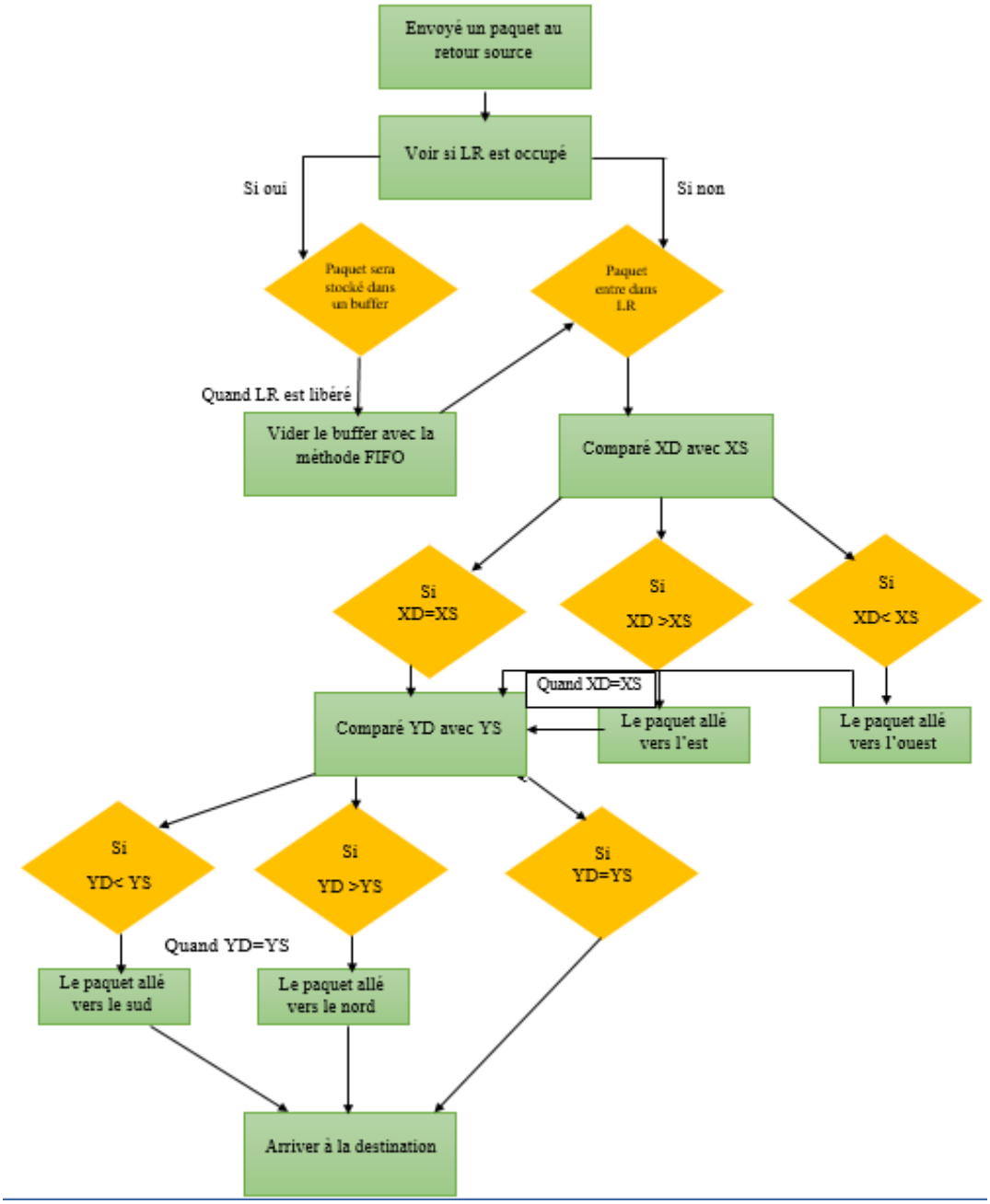


Figure III.5 : diagramme d'acheminement des paquets.

Indication :

XS = X source

XD =X destination

YS = Y source

YD =Y destination

Le diagramme si dessus montre comment l'algorithme XY adaptatif achemine les paquets. Lorsque le paquet arrive au retour source la logique de contrôle vérifie si la logique de

routage est vide donc il n'y a aucun problème le paquet passe directement sinon la logique de contrôle crée un buffer pour stocké les paquets dedans. Lorsque la logique de routage est vide la logique de contrôle commence à envoyer les paquets à la logique de routage et vider son buffer par la méthode FIFO.

Lorsque le paquet entre dans la logique de routage, elle compare le X de l'adresse source avec le X de l'adresse destination, si XD<XS alors le paquet aller vers l'ouest sinon si XD>XS le paquet aller vers l'est, quand on arrive au point ou XD=XS elle compare le Y de l'adresse source avec le Y de l'adresse de destination. Si YD<YS le paquet aller vers le sud sinon si YD>YD le paquet aller vers le nord, quand on arrive au point ou YD=YS donc on est arrivé à la destination.

III.4.3. Formalisations de problème

Un algorithme de routage XY, été proposé au sein du laboratoire LCOMS pour assurer l'arbitrage de routage des paquets d'une source a une destination dans un QNOC. Les travaux précédents ont permis de détecter les limites de cet algorithme dans certaines zones généralement les switchs qui ont 4 ports de sorties et 4 ports d'entrées « N, S, E et O » utilisés pour la communication avec les voisins. On a remarqué que dans cet algorithme les switchs des coins ne fonctionnent pas comme il faut (ils travaillent avec 2 ports au lieu de 4).

III.4.4. Supposition

Dans ce projet de fin d'étude on propose un algorithme de routage adaptatif qui permet de tester la performance et détecter les limites des switchs du QNOC tolérant aux fautes, sachant que l'algorithme proposé par le laboratoire LCOMS est performant sauf certaines situations ne sont pas débloqués ; alors on propose une amélioration de cet algorithme en débloquant ces situations. L'ajout de certaines conditions dans l'algorithme permet d'enlever ce problème en voyant le résultat dans la simulation.

III.5. Conclusion

Dans ce chapitre nous avons commencé par décrire les caractéristiques des Q-Switch, ensuite nous avons parlé sur le commutateur RKT-Switch, après nous avons vu l'algorithme XY proposé par LCOMS et nous avons expliqué sa façon de travailler et ses limites, à la fin nous avons proposé une solution pour améliorer l'algorithme et régler le problème qui s'impose.

Chapitre IV : implémentation, test et résultat

IV.1. Introduction

Après avoir vu toutes les définitions nécessaire et l'étude de cas du switch avec son algorithme de routage, nous allons commencer l'implémentation de notre projet, en premier lieu une présentation du matériel sur lequel nous avons développé notre projet et les différents outils utilisé, ensuite l'exposition des résultats obtenus après les avoir testés par un test Bench et les comparer avec l'algorithme XY avant l'amélioration.

IV.2. Les Ressources matériels

Nous implémenterons notre projet dans un pc portable avec les caractéristiques suivant :

	Pc portable HP
Processeur	Intel Core i7-1065G7
Mémoire	8GB DDR4-SDRAM
Carte graphique	Intel Iris Plus Graphics
Disque dur	256 GO SSD

Tableau IV.1 : caractéristiques de pc.

IV.3. Les outils de développement

IV.3.1. xilinx ISE

Le logiciel XILINX ISE (Integrated Software Environment) est un logiciel multitâche qui possède dans les soft différents outils permettant la création de systèmes ou circuits numériques. D'une manière générale. Le XILINX ISE permet de réaliser toutes les étapes de conception et de programmation des FPGA de XILINX et même pour d'autre circuits programmables tel que les CPLD.

IV.3.2. VHDL

VHDL Est abbreviation de « Very high-speed integrated circuits Hardware Description Language ». L'ambition des concepteurs du langage est de fournir un outil de description

Homogène des circuits, qui permettent de créer des modèles de simulation et de « compiler » le silicium à partir d'un programme unique.

VHDL a été, initialement, conçu comme un langage de simulation, il est fortement marqué par cet héritage très informatique, proche du matériel, qui n'est pas toujours un spécialiste des langages de programmation.

IV.3.3. FPGA

FPGA est un circuit intégré fait pour être reprogrammé par l'utilisateur après sa fabrication en utilisant un langage informatique spécifique sans modifier le matériel (pour plus de détail voire le chapitre (1).

Dans notre projet on a utilisé la carte FPGA XC3S50 de la famille Spartan3 avec le package PQ208.

IV.3.4. Simulateur ISIM

iSim est un Simulateur des circuits logiques et une application web pédagogique d'édition et simulation des circuits logiques combinatoires est séquentiels.

IV.3.5. La bibliothèque

Dans le langage VHDL, on utilise la bibliothèque standard IEEEqui fournit plusieurs package, parmi ces packages on a iee.std_logic qu'on va l'utiliser dans notre travail.

IV.4. Les étapes de travail

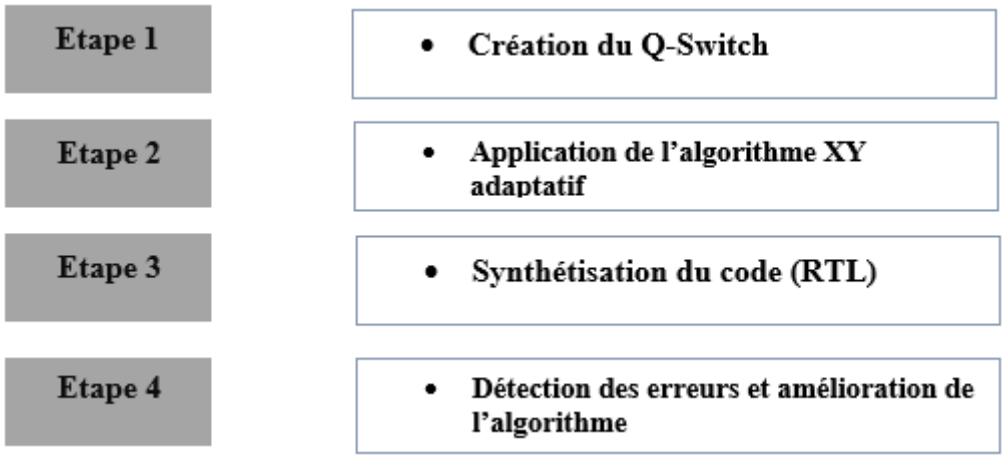


Figure IV.1 : les étapes de travail.

IV.4.1. Création du Q-Switch

Dans cette étape on a un QNOC qui utilise la topologie 2D maillé, nous avons créé un routeur avec les caractéristiques suivantes :

- Il possède 4 ports d'entrées et 4 ports de sortie.

Après avoir synthétisé le code, cette démarche nous donne le schéma du Swicth qu'on a simulé qui est dans le niveau abstrait.

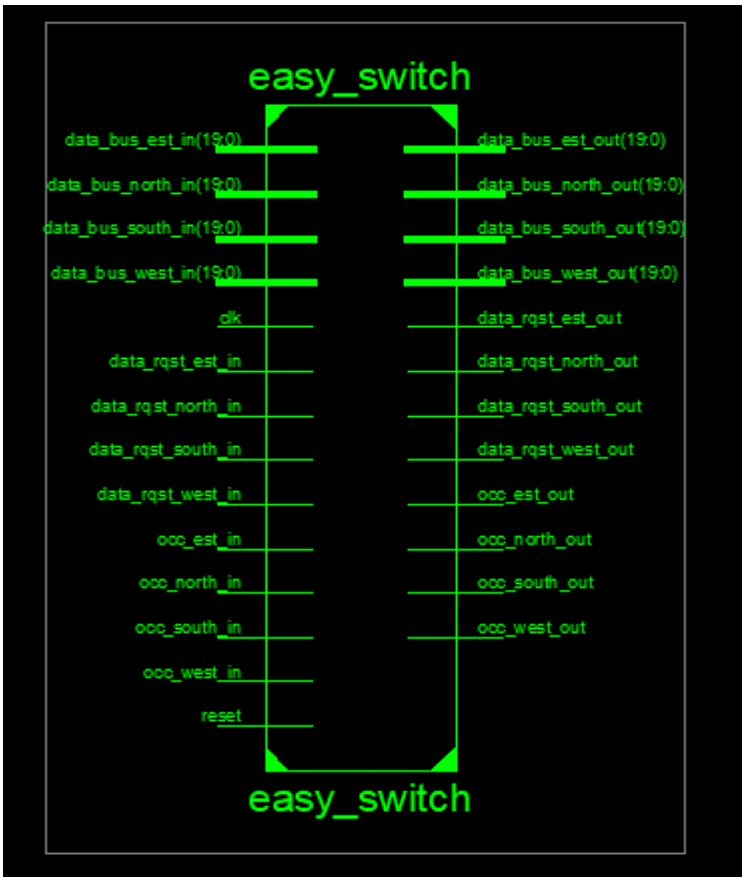


Figure IV.2 : schéma RTL.

RTL (Register Transfer Logic) : Dans la conception de circuits numériques, le niveau de transfert d'enregistrement (RTL) est une abstraction de conception qui modélise un circuit numérique synchrone en termes de flux de signaux numériques (données) entre les registres matériels, et les opérations logiques effectuées sur ces signaux.

Les entrées

data_bus_est_in, data_bus_west_in, data_bus_south_in, data_bus_north_in: des ports réservé pour les bus des données d'entrée qui arrive des 4 directions (nord, sud, est, ouest).

data_rqst_est_in, data_rqst_west_in, data_rqst_north_in, data_rqst_south_in : des ports resérvé pour les requêtes qui indique qu'une données est transmise sur le bus.

occ_est_in, occ_west_in, occ_north_in, occ_south_in: des ports réservé pour indiquer que les voisins sont occupé.

Clk : horloge.

Reset : reset.

Les sorties

data_bus_est_out, data_bus_west_out, data_bus_south_out, data_bus_north_out: des ports réservé pour les bus des données de sortie qui arrive des 4 directions (nord, sud, est, ouest).

data_rqst_est_out, data_rqst_west_out, data_rqst_north_out, data_rqst_south_out : des ports resérvé pour les requêtes qui indique au voisin qu'on lui transmet une donnée.

occ_est_out, occ_west_out, occ_north_out, occ_south_out: des ports réservé pour indiquer que notre routeur est occupé.

- Des bus de données de taille 20 bits, les 4 premiers bits sont réservés pour l'adresse de destination (2 bits pour X et 2 bits pour Y).

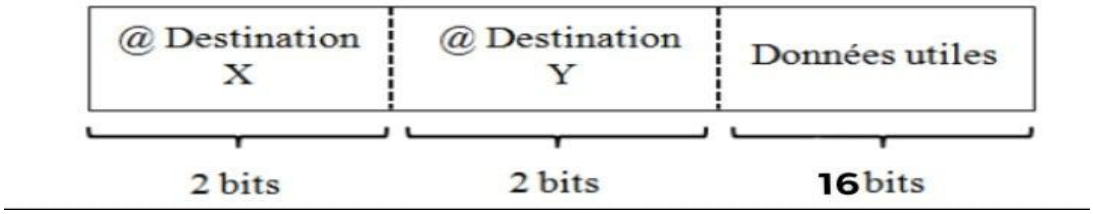


Figure IV.3 : Format de bus de donnée.

Il existe deux blocs de logique dans le routeur sont :

- **Logique de Routage (LR) :**sa place est dans le centre de routeur, elle fait le routage des paquets et elle ne peut router qu'un seul paquet à la fois.

Dans ce bloc on trouve les variables suivantes :

LR_state : un signal de type state_type, ce type est complexe contient 4 paramétrées (ATTENTE_PAQUET, VIDE_PAQUET, LR_OCC, ROUTE) ces paramètres sont expliqués Différentes états de la machine d'états dela logique de routage.

direction_a_router : un signal de type dir_type , ce type est complexe contient 4 paramétrées (EST,WEST,NORTH,SOUTH) , il mémorise la direction qu'il faut router.

reg_lr :un vecteur des signaux de taille n.

LR_occ_signal : un signal provenant de la LR indiquant un état d'occupation à la LC.

- **Logique de Contrôle (LC):** si plusieurspaquets arrive au même temps LC active des signaux qui indique aux routeurs voisins qu'il est occupé et qu'il ne peut pas recevoir temporairement de paquets de données. Si un routeur veut envoyer un paquet il doit attendre que le destinataire soit disponible [64].

Dans ce bloc on trouve les variables suivantes :

State : un signal de type state_type, ce type est complexe contient 4 paramétrées (ATTENTE_PAQUET, VIDE_PAQUET, LR_OCC, ROUTE) ces paramètres sont expliqués Différentes états de la machine d'états de la logique de contrôle et de routage.

rqst_in_vector : un vecteur des signaux de taille 4 bits contenant les requêtes des quartes directions.

data_to_route : un vecteur des signaux de taille n bits contenant les données sortant de la LC et à router.

rqst_to_route : un signal indiquant à la logique de routage qu'une donnée est à router.

reg_paquet_A, reg_paquet_B, reg_paquet_C, reg_paquet_D : un registre de type vecteur de taille n bits stockant les paquets de données arrivant en entré du routeur avant d'être routés.

nb_paquets : un vecteur de type entierde taille 4présente le nombre des paquets qui sont stockés dans le buffer d'entrée.

router_occ_out :un vecteur des signaux qui gère les signaux qui sont occupés dans les sorties.

IV.4.2. Application de l'algorithme XY adaptatif

Il est appelé adaptatif ou tolérant aux fautes car il permet de calculer dynamiquement le chemin emprunté par message.

Notre code est composé de trois parties :

- La première partie est réservée pour la déclaration des variables.
- La deuxième partie est réservée pour la procédure de logique de contrôle, le rôle de cette procédure est de stocké les paquets dans un buffer quand la logique de routage sera occupée, d'où la capacité de buffer est de 4 paquets maximum à la fois.

Si la logique de routage est vide et il y'a un seul paquet en route, le paquet passe directement à la logique de routage, sinon les paquets seront stocké dans le buffer de la logique de contrôle qui utilise la méthode FIFO pour acheminer les paquets (premier entré premier sorti).

Le pseudo code suivant est une partie de cette procédure qui présente comment stocker les paquets si la logique de routage est occupée.

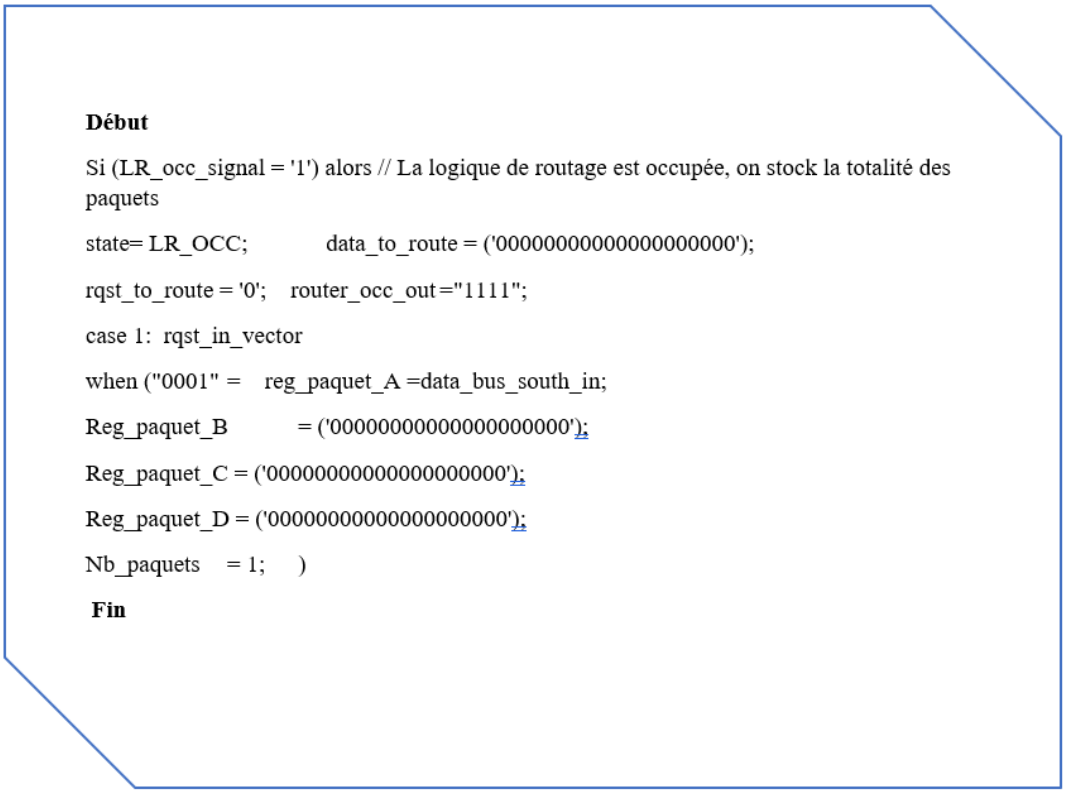
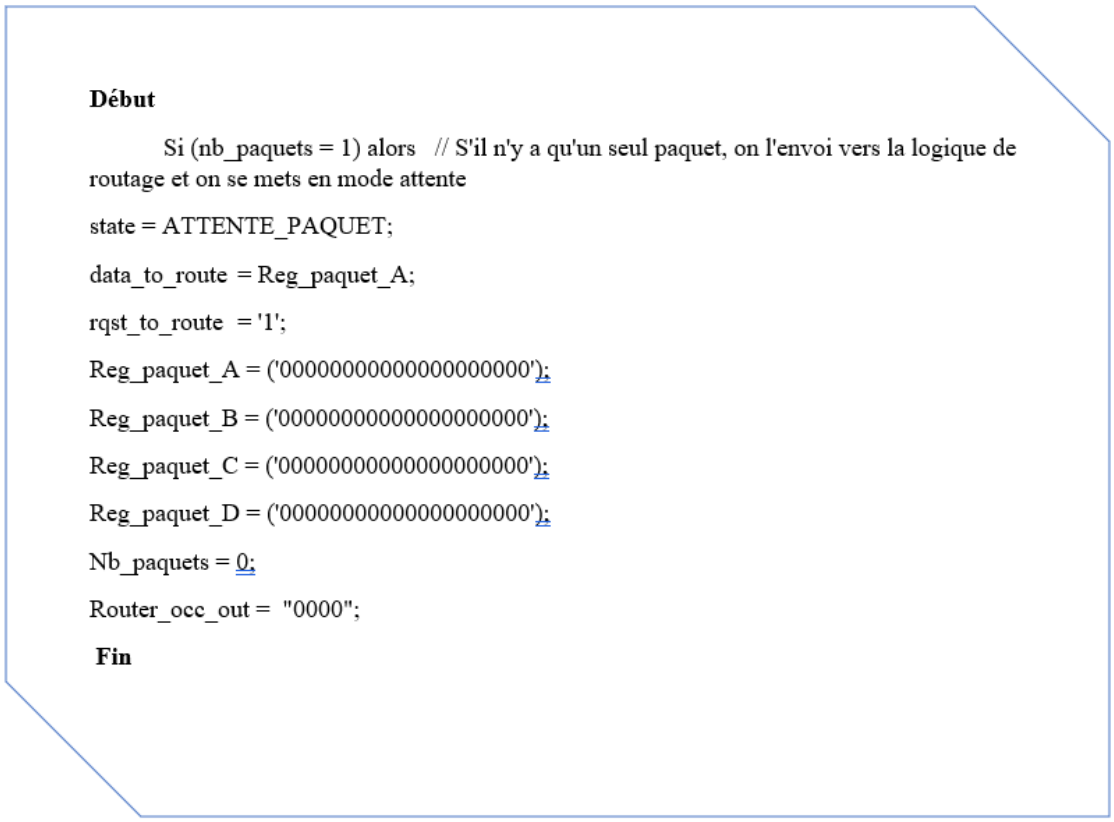


Figure IV.4 : pseudo code 1.

Le pseudo code ci-dessus est présenté comment le buffer de la logique de contrôle envoyé les paques vers la logique de routage.



Début

Si (nb_paquets = 1) alors // S'il n'y a qu'un seul paquet, on l'envoi vers la logique de routage et on se mets en mode attente

state = ATTENTE_PAQUET;

data_to_route = Reg_paquet_A;

rqst_to_route = '1';

Reg_paquet_A = ('00000000000000000000');

Reg_paquet_B = ('00000000000000000000');

Reg_paquet_C = ('00000000000000000000');

Reg_paquet_D = ('00000000000000000000');

Nb_paquets = 0;

Router_occ_out = "0000";

Fin

Figure IV.5 : pseudo code 2.

- La troisième partie et la dernière est réservé pour la procédure de logique de routage, le rôle de cette procédure est d'envoyé le paquet a la destination, elle nous donne le chemin que le paquet doit traverser pour arriver à ça destination.

Pour connaitre le chemin, en premier temps la logique de routage compare le X de la source avec le X de la destination, si XS < XD alors la destination se trouve à l'est, sinon la destination ce trouve à l'ouest. Ensuite elle compare le Y de la source avec le Y de la destination, si YS < YD alors la destination se trouve en nord sinon la destination ce trouve en sud.

Remarque : XS = X source

XD = X destination

YS = Y source

YD = Y destination

S'il fallait router le paquet a une destination précise (soit nord, soit sud, soit est et soit ouest) et cette destination est occupé alors la logique de routage garde le paquet et signaler qu'elle est encore occupée, sinon le paquet sera envoyé directement.

Le pseudo code suivant est une partie de cette procédure qui présente comment choisir le chemin à l'axe x :

```
Début
Si (rqst_to_route = '1') alors
Si (unsigned (data_to_route [3])) > adX alors // Si la destination se trouve à l'est
data_bus_west_out= ('0000000000000000000');   data_rqst_west_out= '0';
data_bus_north_out= ("0000000000000000000') ;     data_rqst_north_out        = '0';
Si (occ_est_in = '1') alors   LR_state = LR_OCC ;
reg_lr  = data_to_route;          direction_a_router  = EST;
data_bus_est_out= ("0000000000000000000');           data_rqst_est_out        = '0';
LR_occ_signal = '1' ;
sinon
LR_state = ROUTE;  reg_lr = ('0');
data_bus_est_out = data_to_route;    data_rqst_est_out= '1';
LR_occ_signal = '0' ;
Fin si ;
data_bus_south_out= ("0000000000000000000');           data_rqst_south_out= '0';
      Fin
```

Figure IV.6 : pseudo code 3.

Apres plusieurs tests de simulation on a constaté que cet algorithme ne marche pas dans les coins, c'est pour cela on a proposé d'ajouter une condition qui débloqué ce problème de façon que les paquets seront envoyer directement au sud pour libérer les buffers et évité toutes les situations d'inter blocage.

Le pseudo code suivant explique comment on a appliqué la condition :

```
Début
        When (autre = LR_state = ROUTE;
data_bus_west_out = '00000000';
 data_rqst_west_out = '0';
data_bus_north_out= '00000000';
data_rqst_north_out= '0';
data_bus_est_out = '00000000';
data_rqst_est_out = '0';
        LR_state = LR_OCC;
        reg_lr  = data_to_route;
        direction_a_router = SOUTH;
        data_bus_south_out = '00000000';
        data_rqst_south_out = '0';
        LR_occ_signal = '0';)
        Fin
```

Figure IV.7 : pseudo code 4.

IV.5. Test et Résultat

IV.5.1. Test

Il existe plusieurs types de testbench comme VOPD[65], MPEG[66] et le testbench aléatoire (Random), mais nous avons choisi de créer notre propre testbench, pour créer un fichier de testbench dans le logiciel ISE, **sélectionnez Projet>** Nouvelle source, et sélectionner **testbenchVHDL** texte dans l'Assistant Nouvelle source. Un vide fichier de relance est ajouté dans le projet. On a défini le testbench dans un éditeur de texte.

IV.5.2. Résultat

Après le lancement de la simulation, les routeurs transmettent les paquets entre eux comme il est illustré dans la figure suivante.

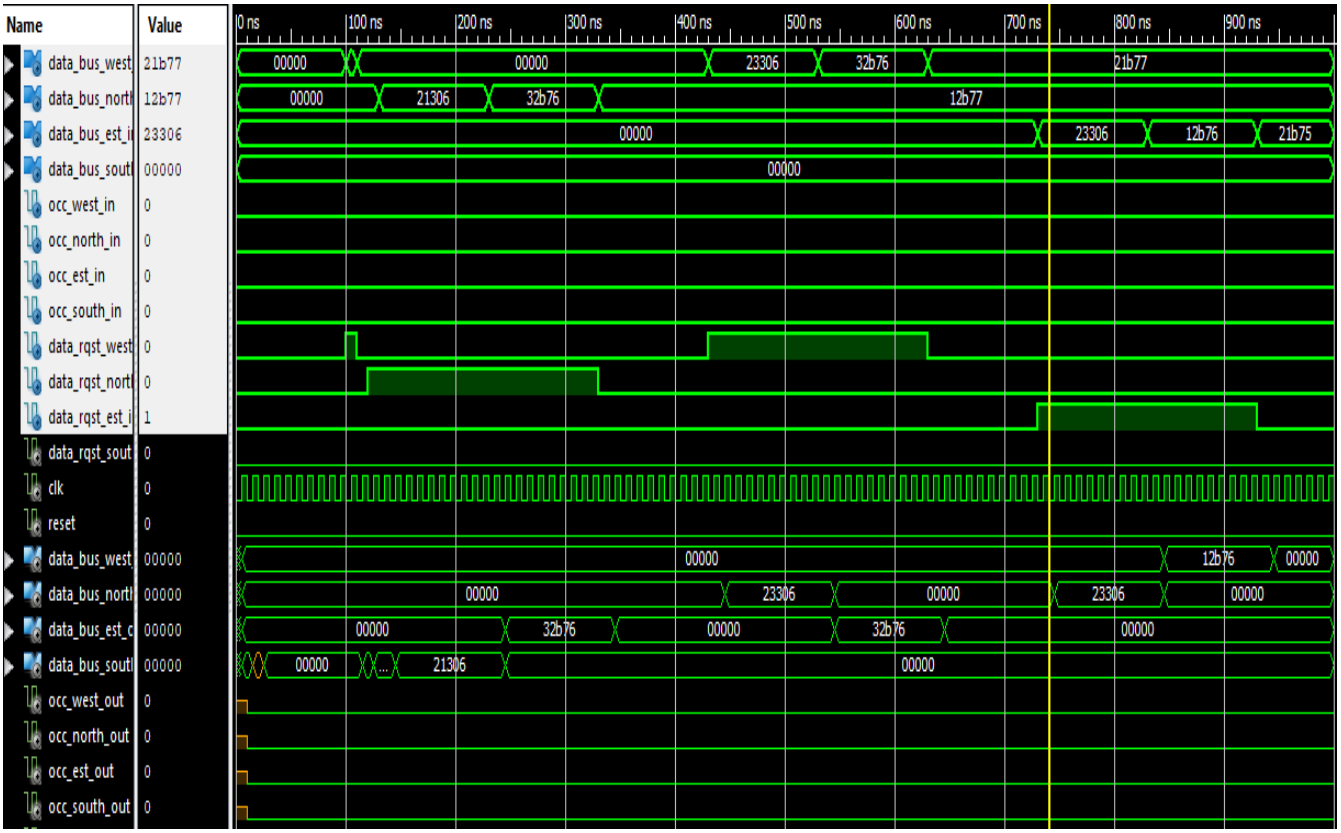


Figure IV.8 : Résultat de la simulation.

L'explication de notre simulation :

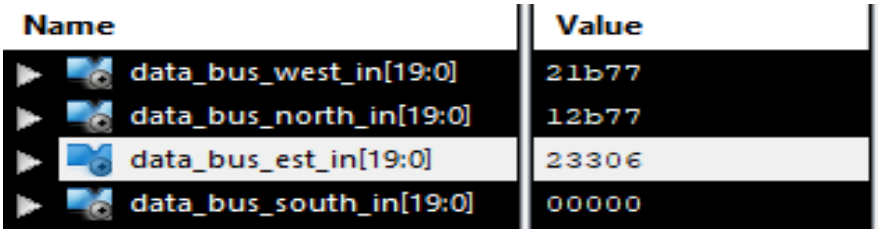


Figure IV.9 : Les bus de données d'entrées.

La figure ci-dessous présente 4 bus de données d'entrés qui ont comme valeur les paquets qu'on voulait acheminer, ce sont des bus que les paquets entre dans la source à travers eux.

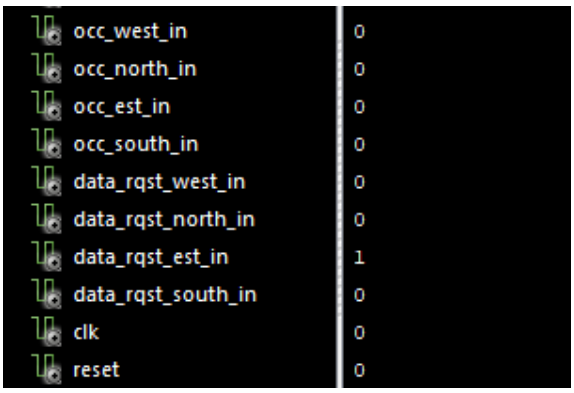


Figure IV.10 : les différents signaux.

La figure ci-dessous présente les différentes valeurs d'état des signaux, d'où 0 est le cas faut (par ex : occ_west_in = 0 veut dire que le port west in n'est pas occupé) et 1 est le cas vrai (par ex : data_rqst_est_in = 1 veut dire qu'on a une donnée transmise sur le bus est).

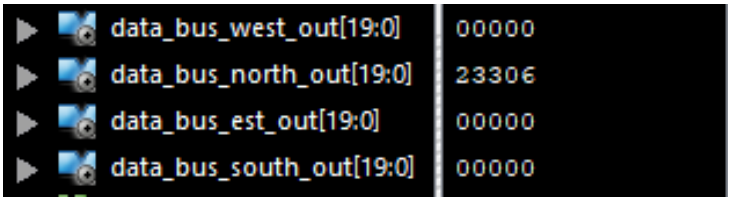


Figure IV.11 : les bus de données de sortie.

La figure ci-dessous présente 4 bus de données de sortie qui ont comme valeur les paquets qu'on voulait acheminer, ce sont des bus que les paquets sort de la source pour arriver à la destination à travers eux.

Maintenant on va suivre le chemin d'un seul paquet, la figure suivante montre comment un paquet est transmis de la source à la destination.

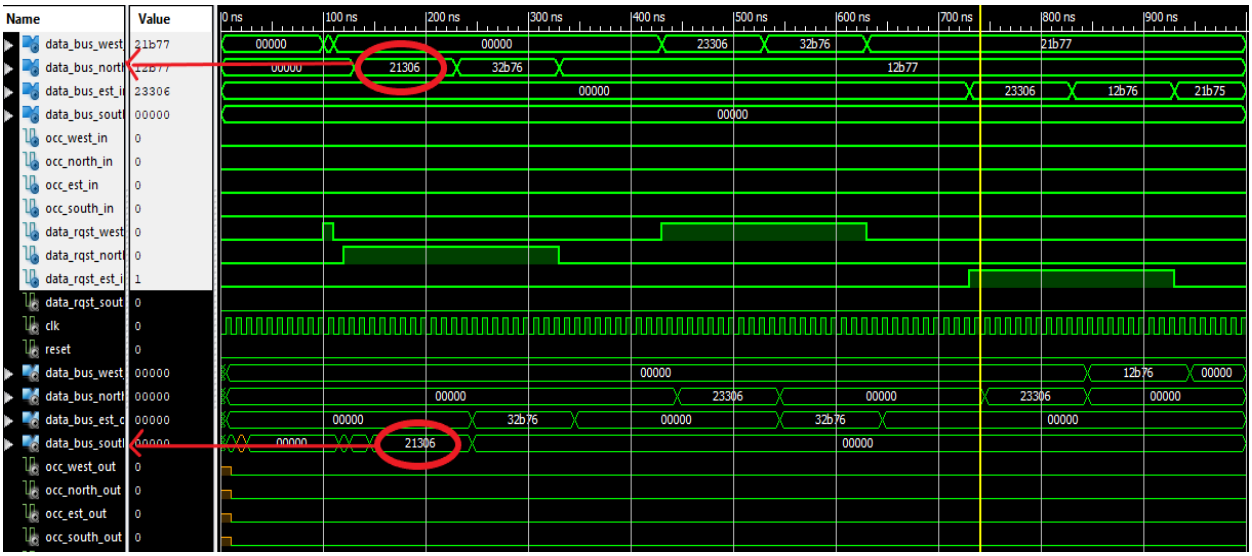


Figure IV.12 : l'acheminement des paquets dans la simulation.

Discussion : après avoir testé le code avec un testbench on a obtenu le résultat de la simulation qui est affiché dans la figure IV.8, on a choisi le paquet 21306 pour suivre ses traces, comme on a déjà parlé les deux premiers bits de paquets sont réserver pour connaitre la destination, dans notre cas le routeur 2.1 est notre destination.

Ce paquet est entré dans le routeur source à travers le port du nord, ensuite il est sorti à travers le port de sud pour arriver à sa destination.

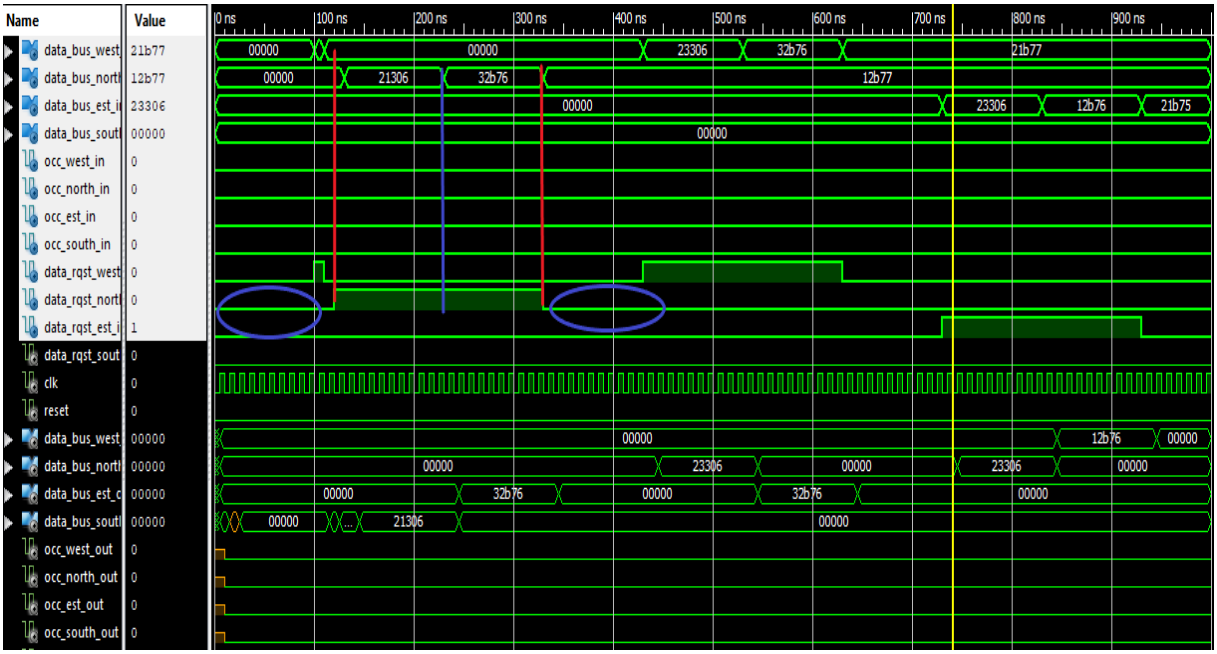


Figure IV.13 : changement d'état.

Discussion : La figure IV.13 présente le changement d'état du signal, il affiche 0 s'il n'y a aucun paquet entraine de transmettre dans le bus de données, et il affiche 1 s'il y a un paquet dans ce bus.

On a choisi le bus de nord pour montrer comment il travail ce signal (data_rqst_north_in), avant 120ns on a aucun paquet donc le signal reste 0 après le paquet 21306 a entré donc le signal saute a 1, dans ce cas on a deux paquets qui sont transmis successivement dans le même bus c'est pour cela le signal reste dans son état 1, dès que le deuxième paquet sorti de ce bus le signal remise à 0 pour attendre un autre paquet.

Le schéma suivant résume comment l'acheminement d'un paquet dans une topologie 2D maillé 4x4.

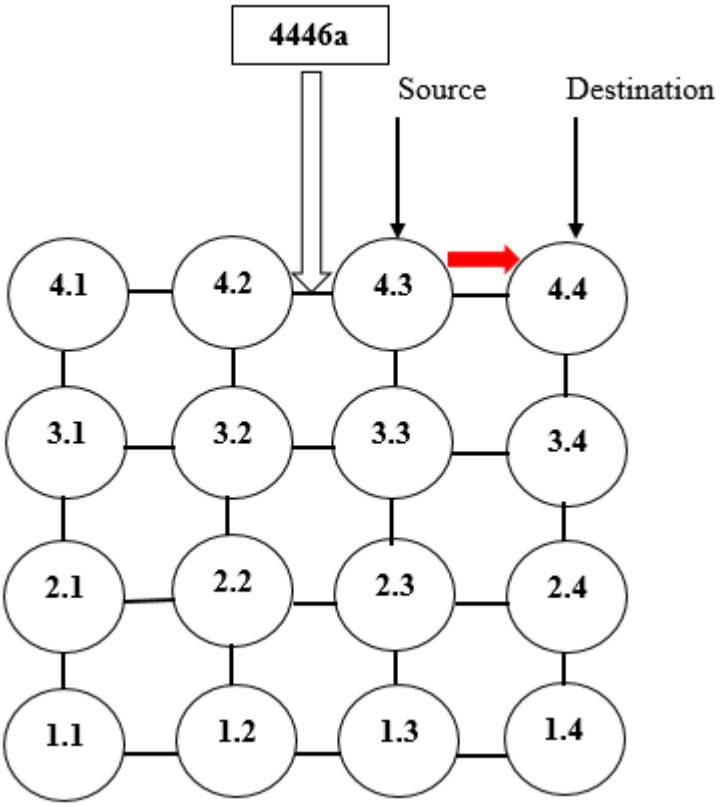


Figure IV.14 : acheminement d'un paquet dans la topologie 4x4.

Comparaison : la différence entre les deux algorithmes est que l'algorithme qu'on a amélioré évite tous les cas d'inter blocage, dans ce cas la performance de l'algorithme amélioré est plus élevée que la performance de l'algorithme avant l'amélioration.

IV.5. Conclusion

Dans ce chapitre on a présenté les différents outils de développement, puis on a expliqué les étapes de notre travail. On a commencé par la création du Q-Switch et on a parlé sur la logique de contrôle et la logique de routage comment elles marchent, ensuite on a vu comment on applique l'algorithme de routage XY adaptatif et on a implémenté notre solution pour effectuer un bon acheminement aux paquets, après on a effectué un testbench pour voir comment l'algorithme XY marche et valider notre proposition.

Notre résultat été une simulation descriptive, elle nous montre le chemin qui prend le paquet pour arriver à sa destination.

On a terminé par faire une petite comparaison entre l'algorithme XY avant et après l'amélioration.

Notre solution prouve son intérêt, les situations d'inter blocage sont réglées dans notre topologie.

CONCLUSION GENERALE & PERSPECTIVES

Les réseaux sur puce est un nouveau paradigme d'interconnexion qui est devenu aujourd'hui une solution privilégiée pour la communication dans les SOC complexes.

Dans ce manuscrit on a choisi les NOC comme solution finale pour connecter différents composants dans une puce, adaptée à la technologie FPGA et on a utilisé l'architecture reconfigurable QNOC qui montre des gains considérables dans les performances avec la topologie MESH à deux dimensions. Le responsable de la communication entre les différents modules interne dans ce paradigme est le routeur, pour cela il utilise des algorithmes de routage qui assure l'acheminement des paquets entre la source et la destination.

Parmi plusieurs algorithmes de routage on a choisi l'algorithme XY adaptatif pour étudier les problèmes d'acheminement des paquets entre les différents nœuds de l'architecture reconfigurable dynamique flexible. Les travaux précédents ont permis de détecter les limites de cet algorithme dans certaines zones c'est pour cela on a eu l'idée de trouver une solution pour régler les problèmes et éviter les cas d'inter-blocage dans la topologie NOC.

Le choix de cette proposition est argumenté par le fait qu'aucune modification dans l'algorithme de routage n'est nécessaire. En outre, notre approche peut être appliquée à tous algorithmes adaptatifs ou tolérants aux pannes de routage basé sur le modèle modifié à leur tour et bien connu algorithme XY.

Notre solution prouve son intérêt. Les situations d'inter-blocage sont éliminées et on a assuré l'acheminement des paquets dans tous les nœuds d'architecture. Ce travail fut une occasion pour nous de complété de manière transversal nos compétence en informatique et d'élargir nos connaissances en XILINX et le langage VHDL.

Comme tout projet de fin d'étude, notre projet n'a pas été réalisé d'une façon parfaite. Ce travail peut connaitre une continuité dans le futur. On pense à une amélioration et application à savoir :

- Valider notre implémentation physique sur le circuit FPGA.
- Faire une étude sur les paquets afin qu'ils soient plus sécurisés durant leurs acheminements.
- Amélioré le routeur qu'on a développé.
- Amélioré la logique de contrôle.

Bibliographies

[1] J.danial (2017 2018). **System-On-Chip Photonic Integrated Circuits** p12,13,21**.**

[2] https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm (15 avril 2022).

[3] https://www.futura-sciences.com/tech/definitions/soc-system-on-chip-14843/ (5 mai 2022).

[4] Elearning-facsci.univ-annaba.dz module ressource cours structure_systeme_micropocessus.

[5] **Les circuits FPGA** : description et applications /GUELLAL Amar.

[6] http://proxacutor.free.fr/architecture.htm (15 mai 2022).

[7] http://proxacutor.free.fr/ (20 mai 2022).

[8] ***Etude des circuits logiques programmable Les FPGA*** /Fabrice CAIGNET/LAAS – CNRS / fcaignet@laas.fr.

[9] U.Meyer-Bäse (mars 1995). ***The use of complex algorithm in the realization of universal sampling receiver using fpgas (in german)***. volume 10, page 215.

[10] https://www.electronicshub.org/introduction-to-asic-technology/ (25 mai 2022).

[11] Ian Kuon and J (Feb 2007). ***Rose. Measuring the gap between fpgas and asics. Computer-Aided Design of Integrated Circuits and Systems***, IEEE Transactions on, 26(2) :203–215.

[12] EETimes. http ://www.eetimes.com/ (25 mai 2022).

[13] Mohamed Amine Boussadi (2007). ***Conception et développement d'un circuit multiprocesseurs en ASIC dédié à une caméra intelligente***, Thèse de doctorat.

[14] ELT5 2021-2022 Vahid MEGHDADI.

[15] Lounis Zerioul (2015) ***Modélisation comportementale d'un réseau sur puce basé sur des interconnexions RF***. Electronique. Université de Cergy Pontoise (UCP).

[16] https://fr.theastrologypage.com/network-chip (30 mai 2022).

[17] Journal of Electrical and Computer Engineering Volume 2012, Article ID 509465,15pages.

[18] DELORME Julien (2007). ***Méthodologie de modélisation et d'exploration d'architecture de réseaux sur puce appliquée aux télécommunications*** . Thèse de doctorat. Institut national des sciences appliquées de Rennes.

[19] Y.A LARIBI (2010). ***Environnement de Mapping pour la Conception des Réseaux sur Puce (NOC)***, thèse d'Ingénieur d'Etat.ESI.

[20] J Delorme (2007). ***Méthodologie de modélisation et d'exploration d'architecture de réseaux sur puce appliquée aux télécommunications***. domain_other.INSAdeRennes.

[21] Wolkotte, P. T. et al. (2005). ***An energy-efficient reconfigurable circuit switched network-on-chip.*** InProceeding sof the19th IEEE international parallel and distribute dprocessing symposium.

[22] A. Mello, L. Tedesco, N. Calazans, and F. Moraes (September 2005). ***Virtual Channels in Net works on Chip***: Implementation and Evaluation on Hermes NoC. In 18th Symposium on Integrated Circuits and Systems Design, pages 178-183, Flo Systems. rianolpolis, Brazil.

[23] Kameche, A. H. (2013). ***Approches basées sur la BBO pour le Data Clustering et le NoCMapping***, Mémoire de magister. Ecole Supérieure d'Informatique ESI, Algérie.

[24] S.Kumar, A.Jantsch, J.Soininen, M.Forsell,M.Millberg, J.Oberg, K.Tiensyrha and A.Hemani, (April 2002). ***A Network on Chip Architecture and Design Methodology***, Proceedings of IEEE Computer Society Annual Symposium on VLSI(ISVLSI), pages 105-112.

[25] W.J. Dally and C.L. Seitz, (1986). ***The Torus Routing Chip,*** Technical Report 5208: TR: 86, Computer Science Dept., California Inst. of Technology, pages 1-19.

[26] Taichi Maekawa (June 1998). ***Survey of NOC Topologies***.

[27] W. J. Dally and B. Towles, (2001). ***Route Packets, Not Wires: On-Chip Interconnection Networks***,Proceedings Design and Automation Conference (DAC), pages 683-689.

[28] P.P.Pande, C.Grecu, A.Ivanov and R.Saleh, (may 2003). ***Design of a Switch for Network on Chip Applications***, Proceedings International Symposium on Circuits and Systems, ISCAS,vol.5, pages 217-220.

[29] P. Guerrier, A. Greiner, (2000). ***A generic architecture for on-chip packet-switched interconnections, in : Design***, Automation and Test in Europe Conference and Exhibition, pp. 250–256.

[30] WJ Dally, B. Towles, (juin 2001). ***Route packets, not wires: on-chip interconnection networks***, DAC, Las Vegas, Nevada, USA.

[31] H. Charlery, E. En renaz, A. Greiner, A. Andriahantenaina, L. Mortiez. (2003). ***SPIN: a Scalable, Packet SwitchedOn chip Micro-network. In Proceedings of the Design***, Automation and Test in Europe Conference and Exhibition (DATE).

[32] A. Greiner, (2005). « The SPIN & DSPIN networks on chip ». Journée Informatique Embarquée : Du matériel au Logiciel.

[33] A. Sheibanyrad, I. Miro Panades, A. Greiner. (juin 2000). ***Systematic Comparison between the Asynchronous and the Multi-Synchronous Implementations of a Network on Chip Architecture***.

[34] F. Karim, A. Nguyen, S. Dey, and R. Rao. (mars 2010). ***On-Chip Communication Architecture for 0C-768 Network Processors***. In Proceedings of the 38th Design Automac.

[35] F. Karim, A. Nguyen, and S. Dey. (September-October 2002). ***An Interconnect Architecture for Networking Systems-on-Chip***. IEEE Micro, 22(5) :36–45.

[36] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. (February 2004). ***QNOC: QoS Architecture and Design Process for Network on Chip***. Journal of System Architecture: The Euromicro Journal, 50(2–3) :105–128.

[37] E. Bolotin, I. Cidon, R. Ginosar et A. Kolodny, (2003). ***Cost considerations in Network on Chip*** . In Special issue on Networks on Chip, Integration - The VLSI Journal.

[38] K. Goossens, J. Dielissen, J. van Meerbergen, P. Poplavko, A. Radulescu, E. Rijpkema, E. Waterlander, and P. Wielage. (2003). ***Guaranteeing The Quality of Services in Networks on Chip. In A. Jantsch and H. Tenchunen, editors, Networks on Chip***, pages 61–82. Kluwer Academic Publisher.

[39] J. Bainbridge and S. Furber. (September-October 2002). ***CHAIN: a Delay-Insensitive Chip Area Interconnect***. IEEE Micro, 22(5) :16–23.

[40] J. Bainbridge et S. Furber, (avril 2003). ***Chain : A Delay-Insensitive Chip Area Interconnect***. IEEE Micro, vol. 22, no5, pages 10–20.

[41] T. Bjerregaard (2005). ***The MANGO Clo kless Network-on-Chip: Con epts and Implementation. Phdcthesis in Informatics and Mathematic al Modelling***, Techni al University of Denmark.

[42] ***Arteries Announces STMicroelectronics Use of NoC for Next Generation Wireless Infrastructure Platform*** . http ://www.arteris.com, March 15 2006.

[43] T. Bjerregaard, J. Sparsø. A S (2005). s***cheduling Discipline for Laten y and Bandwidth Guarantees in Asynchronous Network-on-Chip.*** In Proceedings of the 11th IEEE International Symposium on Asyn hronous Cir uits and Systems, pages 34-43.

[44] R. Dobkin, V. Vishnyakov, E. Friendman, and R. Ginosar. (2005). ***An Asynchronous Router for Multiple Service Levels Networks on Chip***. In Proceedings of the IEEE Int'l Symposium on Asynchronous Circuits and Systems (ASYNC), pages 44–53.

[45] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani, (2002)**.** ***Un réseau sur l'architecture de puce et la méthodologie de conception***, dans : Actes du symposium annuel de l'IEEE Computer Society sur VLSI.

[46] E. Rijpkema, K. Goosens, P. Wielage, (2001). ***Une architecture de routeur pour les réseaux sur silicium***, dans : Actes du progrès, 2e atelier sur les systèmes embarqués.

[47] Article dans Journal de l'Architecture des Systèmes ***·Qos architacture and design process for network on chip*** Février 2004.

[48] WJ Dally, (1987)**.** ***Une architecture VLSI pour les structures de données concurrentes, Kluwer Academic Publishers***.

[49] Slavisa Jovanovic. (2009). ***Architecture reconfigurable de système embarqué auto-organisé. Autre [cs.OH]***. Université Henri Poincaré - Nancy Français.

[50] Evgeny Bolotin, Avinoam Kolodny, Ran Ginosar,(February 2004). Israel ***Cidon.QNoC: QoS architecture and design process for network on chip*** Journal of Systems Architecture .

[51]C. BOBDA, A. AHMADINIA, M. MAJER, J. TEICH, S. FEKETE et J. van der VEEN , (aug 2005). ***Dynoc : A dynamic infrastructure for communication in dynamically reconfugurable devices***. In Field Programmable Logic and Applications. International Conference on, pages 153 – 158, xiv, 94, 96, 106, 107.

[52] S. JOVANOVIC, C. TANOUGAST, S. WEBER et C. BOBDA, (2007). ***A scalable dynamic infrastructure for dynamically reconfigurable systems***. ReCoSoC07. 96, 97, 103, 106, 107, 109, 174.

[53] S. JOVANOVIC, C. TANOUGAST, S. WEBER et C. BOBDA , (2009). ***A new deadlock-free fault-tolerant routing algorithm for nocs***. Field Programmable Logic and Applications. International Conference on, pages 326–331.

[54] CH. Nehnoh, (2018-2019). ***Conception et implémentation d'un algorithme de routage tolérant aux fautes***, Thèse de doctorat université d'Oran.

[55] N.Djaballah (2005-2006) . ***Une architecture d'un réseau sur puce (NoC) dédiée au routage de paquets***, Thèse de doctorat Université de Batna, Magistère Informatique.

[56] L. Benini and G. De Michel. (January 2002). ***Networks on Chips: A New SoC Paradigm***. IEEE Computer Journal, 35(1) :70–78.

[57] L. Benini and G. De Mi heli. (2002). ***Networks on Chip: A New Paradigm for Systems on Chip Design. In Proceedings Design,*** Automation and Test in Europe Conferen e and Exhibition, pages 110-118.

[58] L. Benini and G. De Mi heli. (2002) . ***Networks on Chip: A New Paradigm for Systems on Chip Design. In Proceedings Design***, Automation and Test in Europe Conferen e and Exhibition, pages 418-419.

[59] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. (september2005). ***Efficient Routing in Irregular Topology NoCs***. CCIT Technical Report, 554.

[60] M. Palesi and M.Daneshtalab. (2014). ***Routing Algorithms in Networks-on-Chip***. Springer, New York.

[61] M. Dehyadgari, M. Nickray, A. Afzali-kusha, and Z. Navabi. (December 2005). ***Evaluation of pseudo adaptive XY routing using an object-oriented model for NOC***. International Conference on Microelectronics, pages 5 pp.

[62]: S. JOVANOVIC, C. TANOUGAST et S. WEBER, (2008). ***A new high-performance scalable dynamic interconnection for FPGA-based reconfigurable systems***. ApplicationSpecific Systems, Architectures and Processors. IEEE Conference on, pages 61–66, 2008. 125, 126, 127, 128.

[63] S. Javanovic, C. Tanougast, S.weber. (2002). 19th IEEE int ***Application-specific Architectures apil processors***, 2008. 6861-66.

[64] C. Killian, C. Tanougast, F. Monterio et A. Dandache, (2012) ***.A New Efficient and Reliable Dynamically Reconfigurable Network-on-Chip,*** Journal of Electrical and Computer Engineering, special issue Design and Automation for Integrated Circuits ans Systems,, vol. Article ID 843239, p. 16.

ANNEXE XILINX

Xilinx, Inc. est une société américaine spécialisée dans la **logique** programmable.

Elle conçoit et commercialise des circuits logiques programmables, les outils de développement associés et des services annexes (comme la formation et l'expertise).

Définition

Le logiciel XILINX ISE (Integrated Software Environment) est un environnement de développement qui possède différents outils de CAO. Les sociétés en CAO microélectronique fournissent des environnements logiciels spécialisés. Tous les fabricants de FPGA proposent des outils de CAO pour configurer leurs circuits (XILINX c'est ISE –Fondation pour ALTERA c'est QUARTUS ou MAX+II). C'est un logiciel multitâche qui possède dans ses soft différents outils permettant la création de systèmes ou circuits numériques. D'une manière générale. Le XILINX ISE permet de réaliser toutes les étapes de conception et de programmation des FPGA de XILINX et même pour d'autres circuits programmables talque les CPLD.

XILINX ISE permet :

- la saisie de projets d'implantation (sous forme de schéma logique, de machine d'états ou en langage VHDL, verilog, ABEL, …).
- la simulation du fonctionnement d'un projet d'implantation.
- la transcription en fichier JDEC et l'implantation sur un produit Xilinx.

Historique

Fondation

Xilinx a été fondée en 1984 par trois anciens employés de Zilog : Ross Freeman ; Bernie Vonderschmitt et Jim Barnett, avec comme objectif de commercialiser un composant éléctronique basé sur un nouveau concept : le FPGA.

Bien que située dans la Silicon Valley, la société a choisi de ne pas investir dans sa propre fonderie, mais au contraire de confier l'étape de fabrication de ses composants à des partenaires. Ce modèle de fonctionnement, appelé fabless, c'est largement démocratisé depuis.

Développement

La société a commercialisé son premier produit en 1985, le FPGA XC2064. Deux ans plus tard, elle ouvrait des bureaux de vente en Europe et au Japon. En 1990, Xilinx est introduite en bourse, et réalise en 2000 un chiffre d'affaires dépassant le milliard de Dollars.

Acquisitions

Au cours de son développement, Xilinx a racheté differentes sociétés :

- AccelChip : 13 Janvier 2006
- Hier Design : 7 Juin 2004
- Triscend : 8 Mars 2004
- RocketChips : 3 Octobre 2000
- Veriphia : 21 Septembre 2000
- LavaLogic : 10 Juillet 2000
- Integral Design : 6 Juillet 2000
- Visual Software Solutions : 2 Novembre 2000

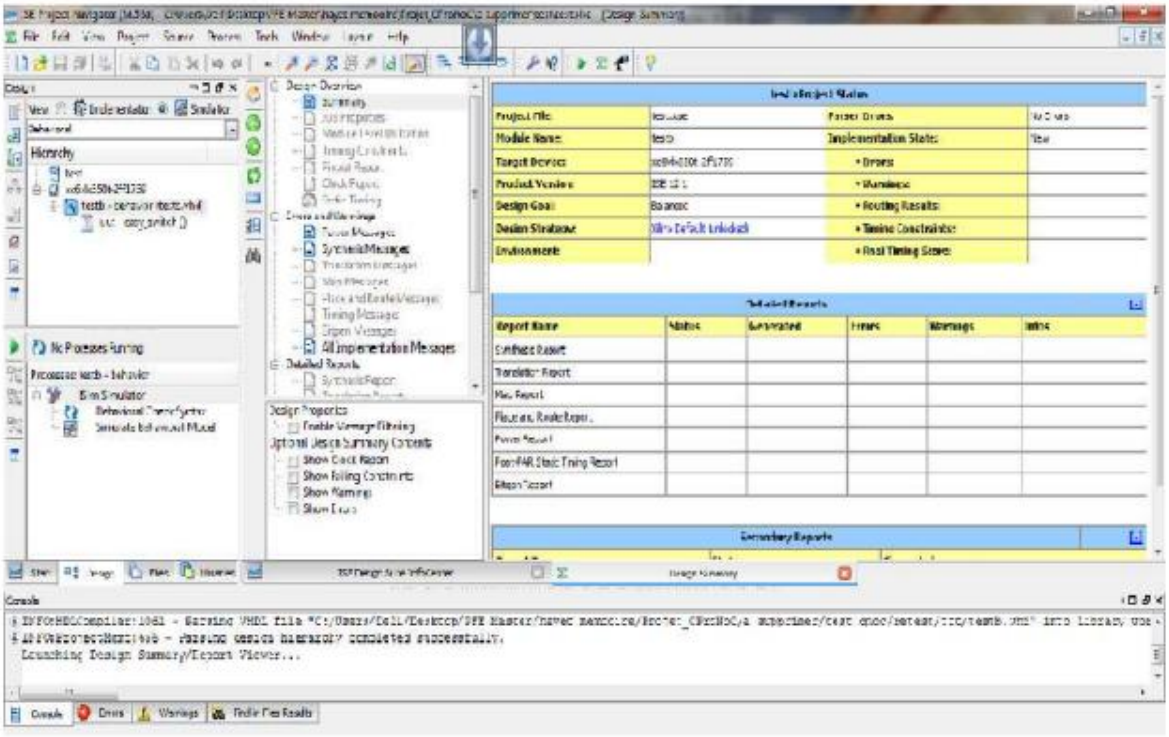


Figure 1 : écran principale de xilinx ISE v12.

Créer un nouveau projet

1- Pour créer un nouveau projet, sur le menu **filechoisissez New project wizard** et donner un nom et une location à ce projet et vous pouvez même faire une description sur ce dernier.

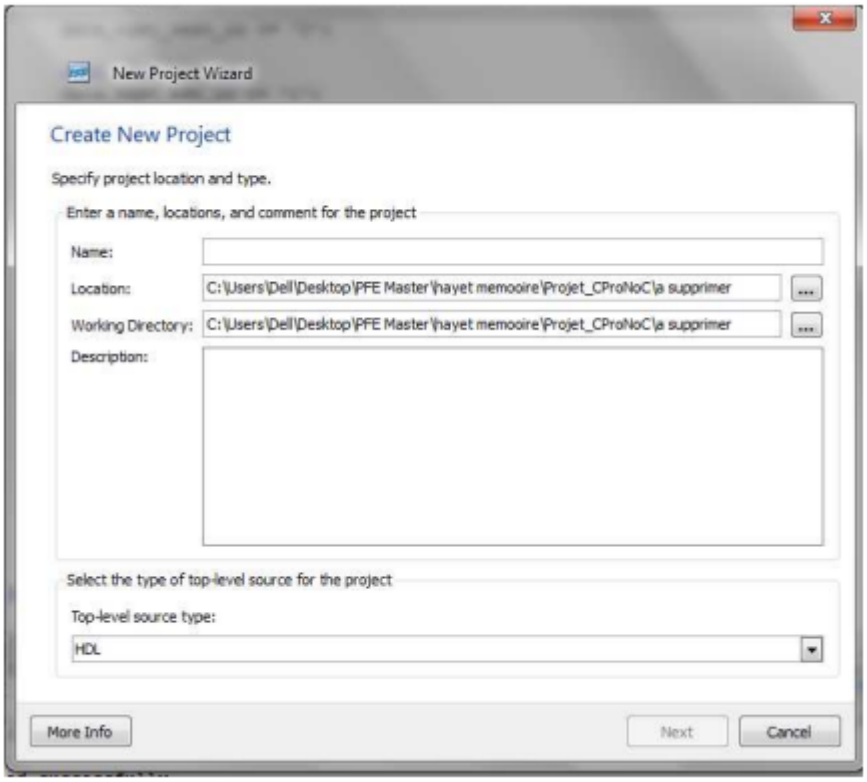


Figure 2 : interface pour créer un nouveau projet.

2- Après vous choisissez des caractéristiques du projet (composant à programmer, outils logiciels).

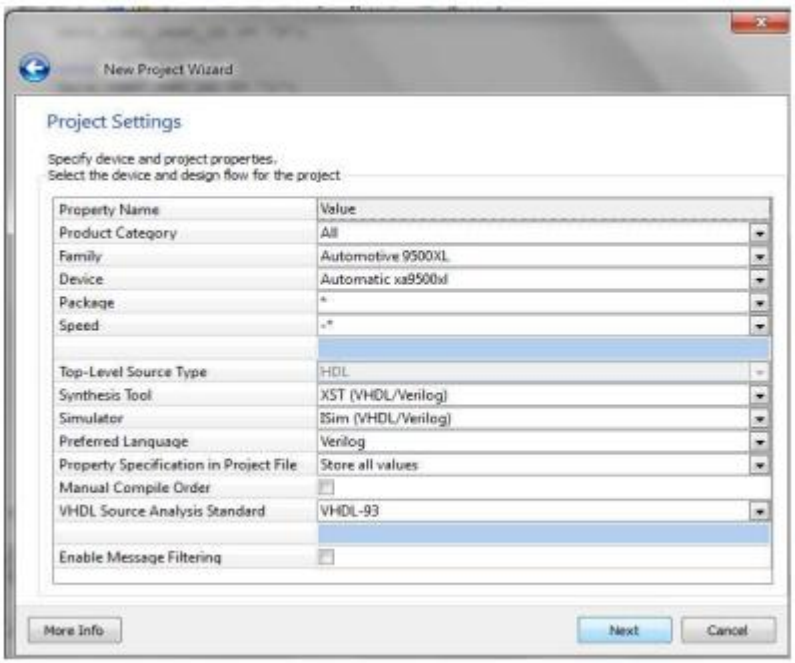


Figure 3 : spécification du simulateur VHDL.

3- Une fois que le projet est ouvert ; on peut apporter du code VHDL ou saisir directement dans le fichier avec extension (.vhd)

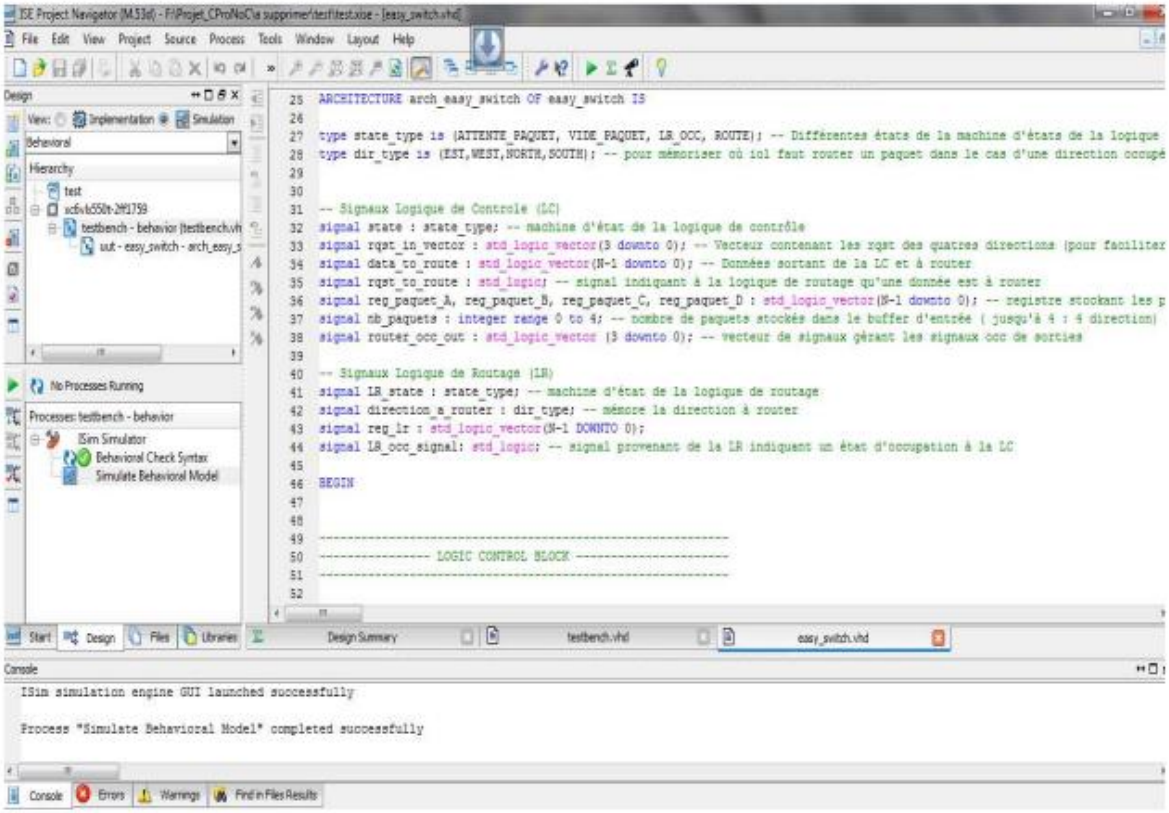


Figure 4 : modèle de VHDL component.

4- Après avoir synthétiser le code ; dans **view implementation** double clic sur **View RTL Schematic / Start**, cette demarche nous donne le schéma du Swicth qu'on a simulé.

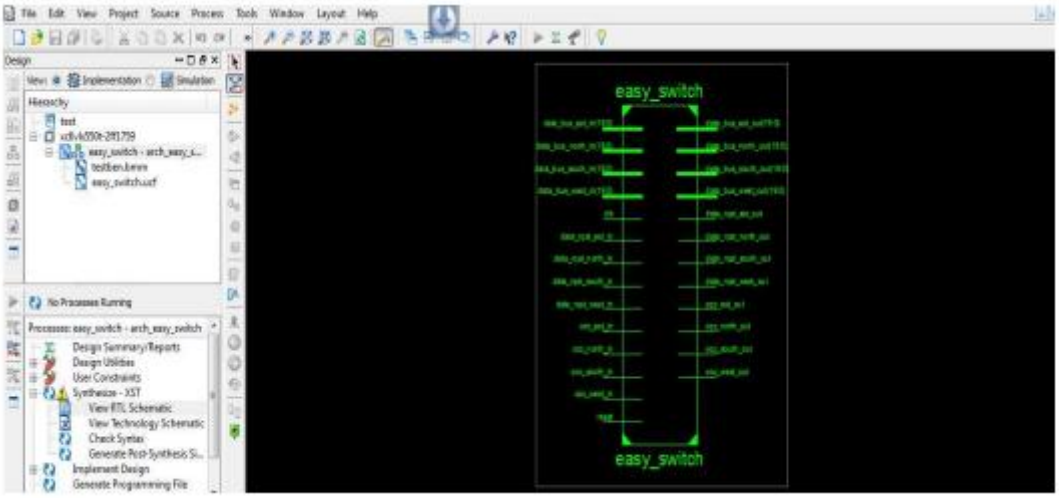


Figure 5 : Vue externe RTL du Easy Q switch.

Avec le double clic sur le schéma vous pouvez voir le niveau le plus bas du Switch

5- Ajout du fichier test bench Pour créer votre propre fichier de st bench dans le logiciel ISE, sélectionnez Projet> Nouvelle source, et sélectionner test bench VHDL texte dans l'Assistant Nouvelle source. Un vide fichier de relance est ajouté à votre projet. Vous

devez définir le st bench dans un éditeur de texte. Simulation VHDL Pour ajouter au banc d'essai tutoriel VHDL pour le projet, effectuez les opérations suivantes:

• Dans Project Navigator, sélectionnez Source Projet> Ajouter.

• Sélectionnez le fichier de test testbench.vhd .

• Cliquez sur Ouvrir.

• Assurez-vous que la simulation est sélectionnée pour le type d'association de fichier.

• Cliquer sur OK

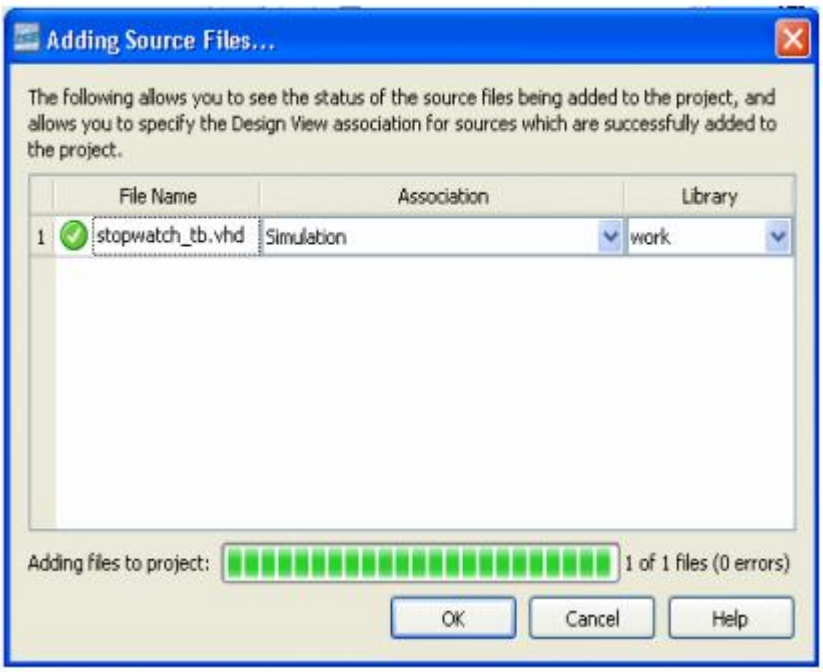


Figure 6 : Fenêtre d'ajout de source.

6- Une fois que vous remplissez dans le banc d'essai (test bench) avec VHDL pour conduire la simulation, vous pouvez vérifier la syntaxe et la simulation de l'onglet processus.

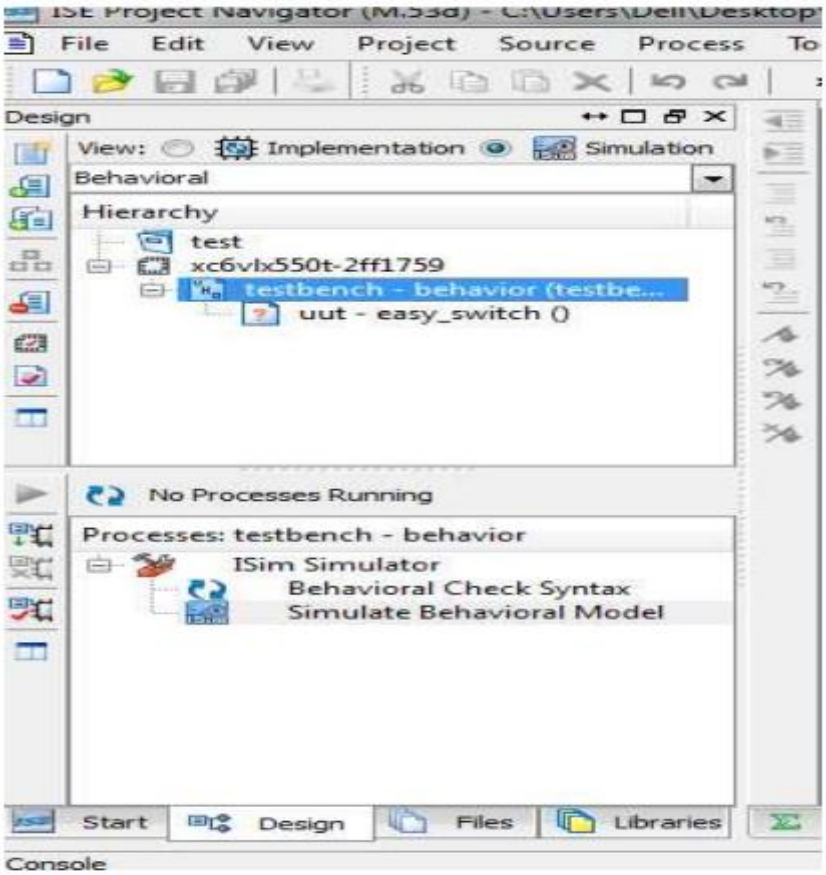


Figure 7 : Lancement de la simulation.

7- Le résultat de la simulation

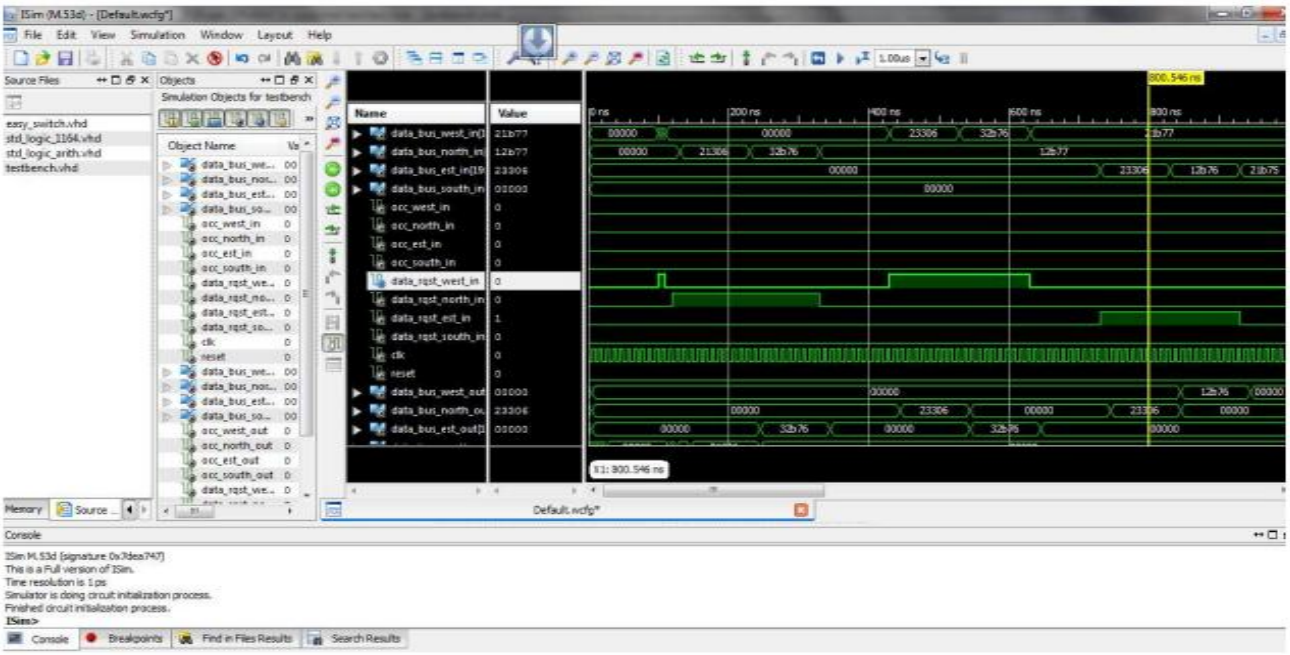


Figure 8 : Résultat de simulation.