

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



UNIVERSITÉ ABOU BEKR BELKAID DE TLEMCEEN

FACULTÉ DE TECHNOLOGIE

DÉPARTEMENT DE TELECOMMUNICATION

MÉMOIRE DE MASTER EN TELECOMMUNICATION

Spécialité : Système De Télécommunications

Présenté par : MEZIANE Amine Djelloul & MAHI Abdelfettah Khaled

Thème

Implémentation matériel d'un système de precodage numérique pour la technologie Massive MIMO

Soutenu publiquement, le 30/09/2021, devant le jury composé de :

Mr. MERIAH Sidi Mohammed	Professeur	Univ. Tlemcen	Président
Mr. BOUSAHLA Miloud	MCB	Univ. Tlemcen	Examineur
Mr. BENDIMERAD Fethi Tarik	Professeur	Univ. Tlemcen	Directeur de mémoire
Mr. BENDIMERAD M. Yassine	MCA	Univ. Bechar	Co-Directeur de mémoire

Année académique : 2020-2021

Remerciements

Avant tout louange à Allah le miséricordieux pour nous avoir donné la force et le courage d'accomplir ce travail. A travers ces quelques lignes, on voudrait exprimé au-delà des remerciements, notre profonde reconnaissance et gratitude envers toutes les personnes ayant contribué de près ou de loin à l'élaboration de ce travail.

En premier lieu, l'expression de nos amples et sincères remerciements ira à notre encadreur Mr **BENDIMERAD** pour son encadrement, sa compétence, sa disponibilité, ses orientations et conseils qui nous ont prodigués durant tout ce travail.

On remercie très chaleureusement, **les membres de jury** qui nous ont fait l'honneur de lire et d'évaluer ce travail.

L'expression de nos profonds respects iront également à tous **les enseignants du département de télécommunication** de Tlemcen pour tous ce qu'ils nous ont inculqué comme connaissance pédagogique durant mon cursus de formation.

C'est une profonde gratitude et sincères mots, qu'on dédie ce modeste travail de fin d'étude à nos chers parents ; qui ont sacrifié leur vie pour notre réussite et qui nous ont éclairé le chemin par leur conseil judicieux Nos remerciements les plus sincères sont adressés à nos frères et sœurs et à notre famille, qui, sans leur aide et soutien, l'accomplissement de ce travail aurait été incertain.

MEZIANE AMINE DJELLOUL

MAHI ABDELFETTAH KHALED

Résumé

Le réseau de communication mobile de cinquième génération (5G) a fait l'objet d'une attention considérable et de nombreuses recherches de la part d'entreprises mondiales, d'instituts de recherche et d'universités. La technique MIMO massive est considérée comme une technique importante dans les 5G, principalement utilisée pour améliorer l'utilisation du spectre et la capacité de canal du système de communication. L'objectif principal de ce travail est l'implémentation d'un système MIMO Massive sur une carte FPGA du type Spartan S3, Nous commençons par la création d'un programme du système MIMO Massive avec beamforming sur le logiciel Matlab pour calculer l'efficacité spectrale en fonction de SNR. Ensuite, nous entamons à générer un programme VHDL du système MIMO Massive à partir de Matlab. Enfin, la simulation de programme sur le logiciel ISE Xilinx qui synthétise l'entité qui a été écrit sous forme VHDL et donne son architecture.

Mots clés

Massive MIMO, VHDL,FPGA,Beamforming ,5G

ملخص

للاتصالات المتنقلة بقدر كبير من الاهتمام والبحث من جانب الشركات العالمية ، (5G) لقد حظيت شبكة الجيل الخامس والتي تستخدم بشكل رئيسي لتحسين استخدام الطيف والقدرة ، G الضخمة من التقنيات المهمة في 5 MIMO وتعتبر تقنية من النوع FPGA الهائل على بطاقة MIMO على توجيه نظام الاتصالات. الهدف الرئيسي من هذا العمل هو تنفيذ نظام لحساب الكفاءة Matlab على برنامج beamforming مع MIMO Mast ونحن نبدأ مع إنشاء برنامج ، Spartan S3 من ماتلاب. وأخيرا ، محاكاة MIMO Mass لنظام VHDL ثم نبدأ في توليد برنامج SNR الطيفية بوصفها وظيفة. ويعطي هيكلها المعمارية VHDL التي توليف الكيان الذي كتب في شكل ISE Xilinx البرنامج على البرنامج

الكلمات المفتاحية

Massive MIMO, VHDL,FPGA,Beamforming ,5G

Abstract

The fifth generation (5G) mobile communication network has received considerable attention and research from global companies, The massive MIMO technique is considered an important technique in 5G, mainly used to improve the spectrum utilization and channel capacity of the communication system. The main objective of this work is the implementation of a Massive MIMO system on a FPGA card of type Spartan S3, We start with the creation of a MIMO Massive program with beamforming on the Matlab software to calculate spectral efficiency as a function of SNR. Then we begin to generate a VHDL program of the MIMO Massive system from Matlab. Finally, the program simulation on the software ISE Xilinx that synthesizes the entity that was written in VHDL form and gives its architecture

Keywords

Massive MIMO, VHDL,FPGA,Beamforming ,5G

Sommaire

Remerciements.....	2
Résumé	3
Sommaire	5
Liste des Figures.....	8
Liste des tableaux	9
Liste des abréviations	9
INTRODUCTION GÉNÉRALE.....	11
CHAPITRE I : SYSTEME DE COMMUNICATION NUMERIQUE.....	13
1.1 Introduction.....	14
1.2 Chaîne de communication	14
1.3 Emetteur.....	15
1.3.1 La Source :.....	15
1.3.2 Le codage de source :	15
1.3.3 Le codage canal :.....	16
1.3.4 Modulateur :.....	16
1.4 Canal de transmission :.....	16
1.4.1 Types de canaux de transmission :.....	17
1.5 Récepteur :	23
1.5.1 Filtre de réception :	23
1.5.2 Estimateur :.....	24
1.5.3 Égalisateur :	24
1.6 Conclusion	26
CHAPITRE II : LES TECHNOLOGIES MIMO ET MASSIVE MIMO	26
2.1 Introduction.....	27
2.2 Types de MIMO :.....	27
2.2.1 Le système SISO (Single-Input Single-Output):	28
2.2.2 Le système MISO (Multiple-input Single-Output).....	29
2.2.3 Le système SIMO (Single-Input Multiple-Output) :.....	29
2.2.4 Le système MIMO (Multiple-input Multiple-Output) :	30
2.3 MODEL DE CANAL MIMO :.....	30
2.3.1 CAPACITÉ D'UN SYSTÈME MIMO :.....	32
2.4 Les codes spatio-temporels	33

2.4.1	Space Time Block Coding (STBC):.....	33
2.4.2	Space Time Treillis Codes (STTC) :.....	34
2.5	Système massive MIMO :.....	35
2.6	Modèle de canal pour mmWave Massive MIMO :.....	36
2.7	Technique de precodage pour les systemes Massive MIMO :	38
2.7.1	Pré-codage numérique:	38
2.7.2	Formation de faisceau analogique:.....	38
2.7.3	Pré-codage hybride :.....	39
2.8	Simulation des systèmes de pré-codage Massive MIMO :.....	40
2.9	Conclusion	41
CHAPITRE III : LES SYSTEMES PROGRAMMABLES FPGA		42
3.1	Introduction.....	43
3.2	Les Réseaux logiques programmables PLD : (PLD : Programmable Logic Device) 43	
3.2.1	Définitions des PLD :.....	43
3.2.2	Les types des PLD :.....	44
3.2.3	FPGA (Field Programmable Gate Array) :.....	45
3.3	La technologie FPGA :.....	46
3.3.1	Structure générale d'un circuit FPGA :.....	46
3.3.2	FPGA de la famille Xilinx :.....	47
3.4	Language VHDL :.....	48
3.4.1	Introduction :.....	48
3.4.2	Entité, Architecture et VHDL Opérateurs :.....	48
3.4.3	Conception de circuits logiques combinatoires et codage simultané en VHDL:52	
3.4.4	Types de données, tableaux et attributs :.....	53
3.4.5	Subtypes	54
3.4.6	Séquentiel Circuit Implémentation in VHDL :.....	54
3.4.7	Packages, Components, Functions et Procedures:.....	55
3.5	Environnement ISE :.....	59
3.6	Conclusion :.....	59
CHAPITRE IV : IMPLANTATION D'UN SYSTEME MASSIVE MIMO SOUS CARTE FPGA.....		60
4.1	Introduction.....	61
4.2	Description de la carte FPGA xilinx spartan S3 :.....	61
4.3	Descriptions du logiciel ISE xilinx : ISE (Integrated Software Environment).....	64
4.4	Description de programme vhdl gener à partir de matlab :	65

4.4.1	Présenter le programme Matlab du massive-MIMO precoding utilisé et l'expliquer le système Massive MIMO simulé	65
4.4.2	Procédure pour générer du code VHDL avec matlab.....	66
4.4.3	Présenter le programme VHDL obtenu	68
4.4.4	Présenter le schéma RTL obtenu (synthétiseur)	69
4.5	Conclusion :	70
CONCLUSION GÉNÉRALE.....		71
Bibliographie.....		72

Liste des Figures

Figure 1.1: structure d'une chaine de communications	14
Figure 1.2:schéma d'un canal de transmission	17
Figure 1.3:description d'un canal binaire symétrique	17
Figure 1.4:diagramme du canal binaire symétrique.	18
Figure1. 5:Modèle du canal AWGN	18
Figure 1.6:Etalement temporel.....	20
Figure 1.7:Effet doppler.....	21
Figure1. 8: Illustration du phénomène de trajets multiples sur le canal radio-mobile	21
Figure 1.9:Canal de Rayleigh.....	21
Figure 1. 10: canal de Rice	22
Figure 1.11: principe de récepteur pour transmission sur onde porteuse	23
Figure 1.12:Illustration de l'égalisation spectrale.	24
Figure 1.13:Illustration des interférences temporelles.	25
Figure 2.1 : <i>schéma général d'un système MIMO</i>	27
Figure 2.2 : schéma du système SISO	28
Figure 2.3 : <i>schéma du système MISO</i>	28
Figure 2.4:schéma du système SIMO.....	29
Figure 2.5:schéma du système MIMO	30
Figure 2.6: model de canal MIMO.....	31
Figure 2.7: schéma de principe du codeur spatio-temporel Alamouti.....	33
Figure 2.8:schéma de principe du décodeur spatio-temporel d'Alamouti	34
Figure 2.9: illustration de MIMO massive et de beamforming.....	35
Figure 2.10 : Massive MIMO	35
Figure 2.11: illustration de rassemblement des informations spatiales dans une matrice	36
Figure 2.12: Formation de faisceau analogique	29
Figure 2.13: Efficacité spectrale avec ZF digital pré-coding en fonction de SNR	40
Figure 2.14:efficacité spectrale des pré-codage MRT,ZF et MMSE en fonction de SNR.	41
Figure 3.1:La structure de base d'un PLD	43
Figure 3.2: Exemple d'une matrice ET	43
Figure 3.3: Exemple d'une matrice OU.....	43
Figure 3.4: la structure simplifiée d'un PA L (2 entrées et 1 sortie)	43
Figure 3.5: Structure générale d'un CPLD	43

Figure 3.6: Structure interne d'un FPGA	43
Figure 3.7:Structure général d'un circuit FPGA.	46
Figure 3.8:processus d'un circuit logique sequentiel	55
Figure 4.1:carte XILINX Spartan -3	62
Figure 4.2:Spartan-3 Family Architecture	64
Figure 4.3:Entité.....	69
Figure 4.4:architecture.....	70

Liste des tableaux

Tableau 2.1 : tableau comparatif entre formation de faisceau analogique, Pré-codage numérique et hybride.	40
--	----

Liste des abréviations

Abréviations	Explications
ASIC	Application-Specific Integrated Circuit
BER	Bit Error Rate
CBS	Canal Binaire Symétrique
CSI	Channel State Information
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device)
CLB	Configurable Logic Bloc
DSP	Digital Signal Processor
DSD	<i>Power Spectral Density</i>
DDR	Double Data Rate
DCI	Digitally Controlled Impedance
DCM	Digital Clock Manager
EPLD	Erasable Programmable Logic Device
EECMOS	Electrically Erasable Complementary Metal Oxide Semiconductor
EDK	Embedded Development Kit

FPGA	Field Programmable gate Arrayou
GAL	Generic Array Logic
HDL	Hardware Description Language
IES	Interférence Sntre symboles
IOB	Input Output Bloc
ISE	Integrated Synthesis Environment
IC	Circuits Intégrés
LOS	Line Of Sight
LVDS	low-voltage differential signaling
LUT	Look-up Table
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-input Single-Output
MMSE	Erreur Quadratique Moyenne Minimale
NLOS	Non Line Of Sight
OFDMA	Orthogonal Frequency-devision Multiple Access
PLL	Phase-Locked Loop
PAL	Programmable Array Logique
PLD	Programmable Logic Device
RF	Radio Frequency
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RSDS	Reduced Swing Differential Signaling
SISO	Single-Input Single-Output
SIMO	Single-Input Multiple-Output
SNR	Signal-to-Toise Ratio
STBC	Space Time Block Coding
STTC	Space Time Treillis Codes
SRAM	Static Random Access Memory
SDRAM	Synchronous Dynamic Random Access Memory
SDK	Software Development kit
TEB	Taux d'Erreur par Bit
ULA	Uniform Linear Array

UPA	Uniform Planar Array
UV	Ultra-Violet
VHDL	Very High speed integrated circuit hardware Description Language
ZF	Forçage à Zéro

INTRODUCTION GÉNÉRALE

Le développement rapide de la communication mobile sans fil a entraîné une croissance explosive du nombre d'utilisateurs mobiles et de l'ampleur des industries connexes. Par conséquent, le système de communication sans fil doit répondre à un taux de transmission de données plus élevé et à une capacité du système plus élevée, et le système de communication doit utiliser efficacement la ressource de bande passante. En raison de la pénurie de ressources spectrales, il est très important d'améliorer l'utilisation du spectre du système.

La technique MIMO peut utiliser des ressources spectrales limitées pour répondre à la demande des utilisateurs en matière de performances du système. Mais il y a moins d'antennes dans la station de base du système MIMO traditionnel et les performances du système sont limitées. En tant que l'un des techniques clés de la future 5G, le MIMO massif peut répondre aux besoins des futures entreprises de communication sans fil, améliorer l'efficacité du spectre et la capacité des canaux du système de communication, et améliorer efficacement la fiabilité des liaisons et le taux de transmission des données.

La technique MIMO a été proposée pour la première fois par Marconi en 1908. Elle est équipée de plusieurs antennes à la fois à l'émission et à la réception pour améliorer la capacité du système de communication, le taux de transmission des données du système et la fiabilité de la transmission. 3 GPP LTE Release10 peut déjà prendre en charge 8 ports d'antenne pour la transmission, c'est-à-dire 8 utilisateurs à flux unique ou 4 utilisateurs à double flux pour une transmission simultanée.

En 2010, Marzetta, un scientifique du laboratoire Bell, a proposé le concept de MIMO massif dans le contexte d'un scénario à cellules multiples et TDD. Ainsi, certaines caractéristiques

différentes du nombre limité d'antennes dans une seule cellule ont été trouvées. La technique MIMO massive fait référence au fait que la station de base est équipée d'un grand nombre d'antennes, généralement une centaine ou plusieurs centaines d'antennes, ce qui est plusieurs ordres de grandeur supérieurs au nombre d'antennes du système de communication existante. Il dessert plusieurs utilisateurs simultanément sur la même ressource temps-fréquence, et les terminaux mobiles adoptent généralement le mode de communication de la réception d'une seule antenne.

L'objectif de notre travail est l'implémentation d'un système MIMO massive sur une carte FPGA. Cette dernière est un circuit intégré à grande échelle qui peut être programmée après sa fabrication plutôt que d'être limité à une fonction matérielle prédéterminée et immuable. La technique FPGA est largement utilisée dans les systèmes de communication sans fil. Il combine la vitesse du matériel dédié et optimisé pour les applications et la reprogrammable des microprocesseurs, ce qui le rend adapté à la mise en œuvre à grande vitesse de filtres adaptatifs.

CHAPITRE I : SYSTEME DE COMMUNICATION NUMERIQUE

1.1 Introduction

Un système de communication est un système qui permet l'échange des informations à distance depuis la source jusqu'à un utilisateur à travers un canal. Dans les systèmes de radiocommunication mobiles, le signal transmis passe par un canal à trajets multiples avant d'arriver au récepteur. Ce canal radio a deux caractéristiques physiques importantes de son comportement : la sélectivité en fréquence et la sélectivité en temps. Pour lutter contre les sélectivités fréquentielles et temporelles, les récepteurs classiques sont en général formés par la mise en cascade de plusieurs modules disjoints : égaliseur, décodeur, démodulateur, etc. Ou chaque module réalise son traitement indépendamment des autres modules. Un tel concept est le sou optimal et ne conduit pas aux performances optimales à cause de la perte d'information.

Ce chapitre décrit brièvement les différentes parties d'un système de communication numérique. D'abord, nous décrirons le fonctionnement d'une chaîne de transmission numérique, de la source d'information binaire au destinataire, par les étapes successives de codage, de modulation, de transmission dans un canal physique et de démodulation, Ensuite, nous présenterons les différents types des canaux puis le récepteur avec ses paramètres.

1.2 Chaîne de communication

Les systèmes de transmission numérique véhiculent de l'information sous forme numérique entre une source et un ou plusieurs destinataires en utilisant un support physique comme le câble, la fibre optique ou encore la propagation sur un canal radioélectrique [1]. Les signaux transportés peuvent être soit directement d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche du système de transmission est d'acheminer l'information de la source vers le destinataire avec le plus de fiabilité possible. Les caractéristiques de l'environnement de transmission sont très importantes et affectent directement la conception des systèmes de communication et leurs fonctions.

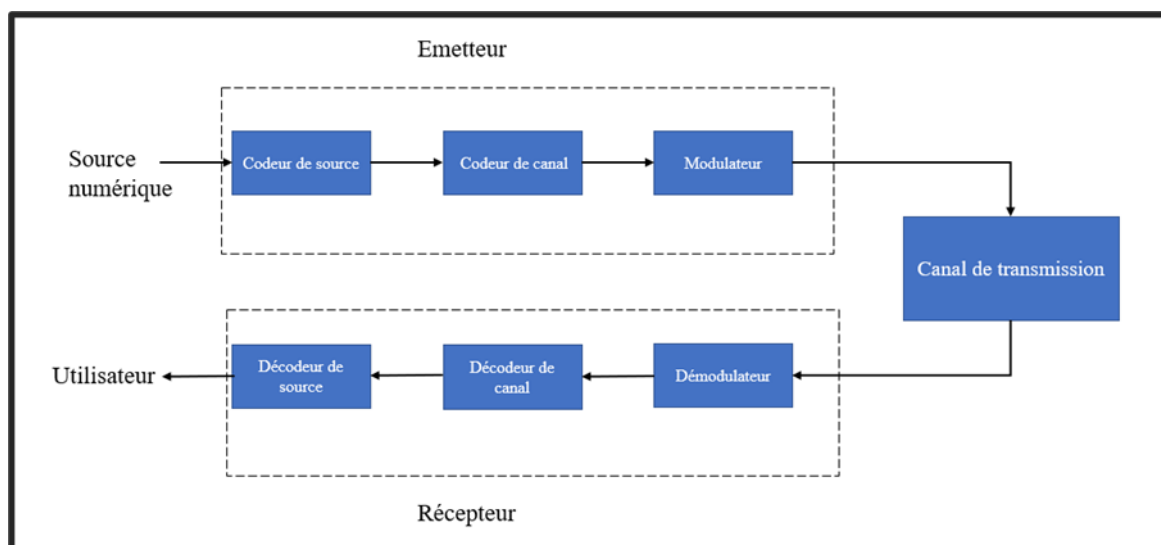


Figure 1.1: structure d'une chaîne de communications

Le schéma de principe d'une chaîne de transmission numérique est représenté sur la figure (1.1). On peut distinguer : la source de message, le milieu de transmission et le destinataire qui sont des données du problème le codage et le décodage de source, le codage et le décodage de canal, l'émetteur et le récepteur représentent les degrés de liberté du concepteur pour réaliser le système de transmission. On décrit de façon succincte les différents éléments qui constituent une chaîne de transmission en partant de la source vers le destinataire.

Les trois caractéristiques principales permettant de comparer entre les différentes techniques de transmission sont les suivantes :

- la probabilité d'erreur par bit transmis permet d'évaluer la qualité d'un système de transmission. Elle est fonction de la technique de transmission utilisée, mais aussi du canal sur lequel le signal est transmis. En pratique, elle est estimée par le Taux d'Erreur par Bit TEB.
- l'occupation spectrale du signal émis doit être connue pour utiliser efficacement la Bande passante du canal de transmission. On est contraint d'utiliser de plus en plus des Modulations à grande efficacité spectrale.
- la complexité du récepteur est le troisième aspect important d'un système de transmission [2].

1.3 Emetteur

Le message numérique, en tant que suite d'éléments binaires, est une grandeur abstraite, pour transmettre ce message, il est donc nécessaire de lui associer une représentation physique, sous forme d'un signal. C'est la première fonction de l'émetteur, appelée généralement, opération de "modulation", le type des signaux dépend bien entendu des propriétés physiques du milieu de transmission que le signal va traverser, et des exigences (TEB) de la liaison [2].

1.3.1 La Source :

La source d'information est le premier maillon de la chaîne de transmission. Elle fournit le message porteur de l'information. Le message à transmettre doit être sous forme numérique. Si la source délivre un message analogique (parole en sortie d'un microphone, image en sortie d'une caméra...), il faut le numériser. Cette numérisation s'effectue en échantillonnant le message analogique et en quantifiant les échantillons obtenus. Chaque échantillon quantifié est ensuite codé sur les éléments binaires appelés bits [2].

1.3.2 Le codage de source :

Consiste à supprimer la redondance contenue dans les messages de la source de l'information. Il peut être avec ou sans pertes d'information. La compression avec pertes vise les signaux numérisés (image, audio ou vidéo) [3].

Après numérisation et codage, la source de message numérique est caractérisée par son débit binaire D . Ce dernier défini comme le nombre d'éléments binaires émet par unité de temps.

D est égal à :

$$D = \frac{1}{T_b} \text{ (bit/s)} \quad (1.1)$$

Avec T_b est l'intervalle de temps séparant l'émission par la source de deux éléments binaires.

1.3.3 Le codage canal :

Le codage canal, aussi appelé codage détecteur et/ou correcteur d'erreurs, est une fonction spécifique des transmissions numériques, il introduit la redondance dans la séquence d'information, cette opération conduite à une augmentation du débit binaire de la transmission, le décodeur de canal qui connaît la loi de codage utilisée à l'émission, donc il est capable de détecter et corriger les erreurs de transmission [3].

La fonction de codage de canal n'est pas toujours utilisée, car elle accroît la complexité des équipements de transmission et donc leur coût [4].

Différents types de codes ont été utilisés jusqu'à présent, parmi lequel on distingue : les codes en blocs et les codes convolutifs.

- **Les codes en blocs** : l'information est d'abord coupée en bloc de taille constante et chaque bloc est transmis indépendamment des autres, avec une redondance qui lui est propre.
- **Les codes convolutifs** : consistent à découper le message en blocs finis, et le considérer comme une séquence semi-infinie $a_0a_1a_2\dots$ des symboles qui passent à travers une succession de registres à décalage, dont le nombre est appelé mémoire du code.

1.3.4 Modulateur :

Il change généralement la forme physique du signal et peut être modulateur de fréquence, d'amplitude ou de phase, numérique ou analogique.

1.4 Canal de transmission :

Le rôle de tout système de communication est d'assurer que le récepteur comprenne l'intégralité des messages transmis par l'émetteur, quel que soit la compression, le format ou le type des données, mais aussi les perturbations induites sur le canal de transmission et son effet parasite [5]. La figure suivante présente un schéma général un canal de transmission

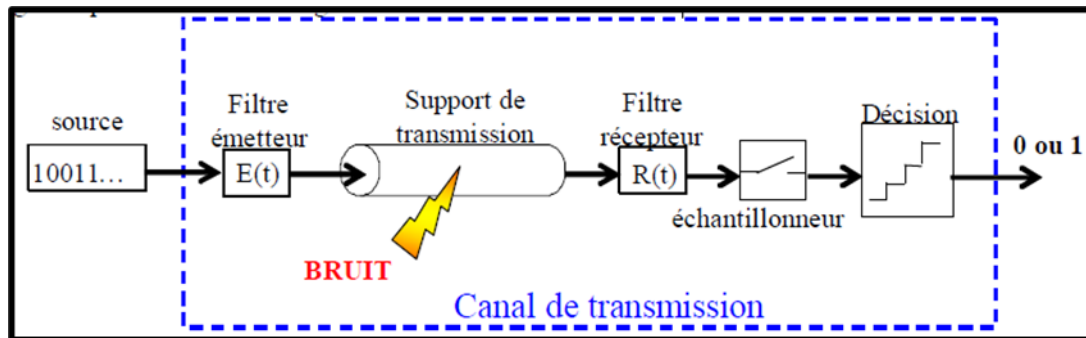


Figure 1.2:schéma d'un canal de transmission

Le transfert de l'information nécessite une source de données, traduites dans un système compréhensible par l'émetteur et le récepteur (codage, format, compression préalablement définie). Le canal proprement dit représente le lien ou le support de transport de l'information entre les 2 entités communicantes, mais il comprend aussi les dispositifs en entrée et en sortie du support de transmission qui va aider à l'émission, à la réception et à l'extraction correcte des données numériques. Pour envoyer le signal à travers le canal, la source a besoin d'un système d'adaptation (physique pour mettre en forme le signal, logiciel pour le protocole de dialogue). Le signal peut être directement transmis à travers le canal, la transmission se fait alors sur la même bande de fréquences que le signal à transmettre.

1.4.1 Types de canaux de transmission :

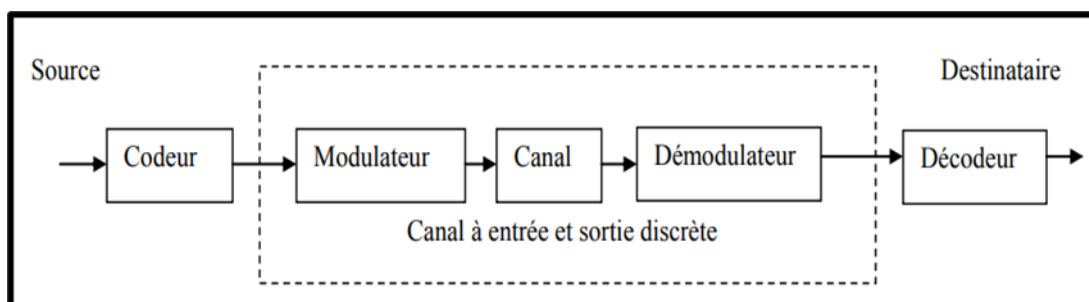


Figure 1.3:description d'un canal binaire symétrique

1.4.1.1 Le canal binaire symétrique (CBS) :

Le canal binaire symétrique (CBS) est le modèle le plus simple des canaux de transmission, c'est un canal discret dont les alphabets d'entrée et de sortie sont finis et égaux à $\{0,1\}$. On considère dans ce cas que le canal comprend tous les éléments de la chaîne compris entre le codeur de canal et le décodeur correspondant.

On note respectivement a_k et y_k les éléments à l'entrée et à la sortie du CBS. Si le bruit et autres perturbations causent des erreurs statistiquement indépendantes dans la séquence binaire transmise avec une probabilité p , alors :

$$prob(y_k = 0|a_k = 1) = prob((y_k = 1|a_k = 0) = p \quad (1.2)$$

$$prob(y_k = 0|a_k = 0) = prob((y_k = 1|a_k = 1) = p - 1 \quad (1.3)$$

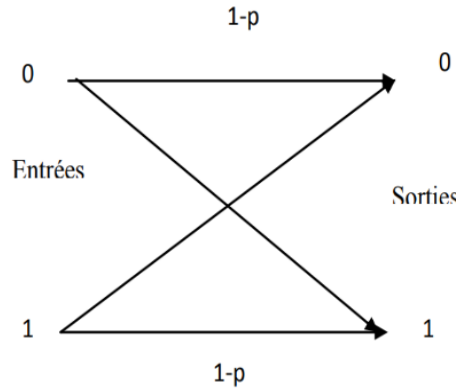


Figure 1.4:diagramme du canal binaire symétrique.

Chaque élément binaire à la sortie du canal ne dépendant que de l'élément binaire entrant correspond, dans ce cas le canal est dit « **sans mémoire** » [6].

1.4.1.2 Canal à bruit additif gaussien blanc AWGN : (Average White Gaussian Noise) :

Un modèle de canal largement utilisé dans l'analyse des systèmes de communication est le bruit blanc gaussien additif (AWGN), ou plus simplement canal avec bruit gaussien. Dans ce modèle, la puissance du bruit a une densité spectrale uniforme (bruit blanc), qui s'ajoute au signal d'origine. La distribution de bruit résultante est un processus gaussien avec une moyenne nulle [1]. Ce n'est peut-être pas toujours très réaliste, mais cela simplifie grandement les calculs associés à l'estimation des performances d'un système de communication. En fait, la plupart des courbes de taux d'erreur (BER) sont générées en supposant que le canal est un bruit gaussien. Le signal reçu s'écrit alors :

$$r(t)=s(t)+v(t) \quad (1.4)$$

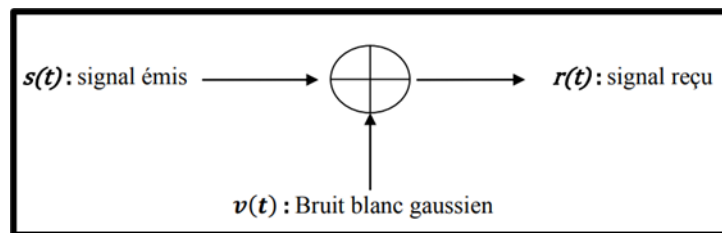


Figure1. 5:Modèle du canal AWGN

Où $v(t)$ représente le bruit, caractérisé par un processus aléatoire gaussien de moyenne nulle, de variance σ_n^2 et de densité spectrale de puissance bilatérale $\Phi_n = \frac{N_0}{2}$. La densité de la probabilité conditionnelle de $r(t)$ est donnée par l'expression :

$$f(r/s) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{(r-s)^2}{2\sigma_v^2}} \quad (1.5)$$

1.4.1.3 Canaux à évanouissements :

Les communications radio ont souvent besoin d'un modèle plus élaboré prenant en compte les différences de propagation du milieu, appelées encore atténuations ou évanouissements, qui affectent la puissance du signal. Cette atténuation du signal est principalement due à un environnement de propagation riche en échos et donc caractérisé par de nombreux multi trajets, mais aussi au mouvement relatif de l'émetteur et du récepteur entraînant des variations temporelles du canal.

Le phénomène de multi-trajets s'observe lorsque l'onde électromagnétique portant le signal modulé se propage par plusieurs chemins de l'émetteur au récepteur. En ce qui concerne les variations temporelles du canal, on peut distinguer deux classes, l'étalement temporel et l'effet Doppler, pouvant par ailleurs constituer une base pour la classification des canaux à évanouissements [1].

1.4.1.3.1 Etalement temporel :

Lors d'une transmission sur un canal à évanouissements, les composantes du signal ayant emprunte des chemins distincts arrivant au récepteur avec des retards différents. L'étalement temporel, note et défini par la différence entre le plus grand et le plus court des retards, permet de caractériser le canal par une seule variable qui est la dispersion temporelle du canal. La bande de cohérence du canal, notée correspond à la gamme de fréquence sur laquelle les amplitudes des composantes fréquentielles du signal, fortement corrélées, subissent des atténuations semblables. En dehors de cette bande de fréquence en revanche, les distorsions du signal deviennent non négligeables [7].

En général, la bande de cohérence d'un canal est du même ordre de grandeur que l'inverse de son étalement temporel :

$$B_c = \frac{1}{T_m}, \quad B_s \text{ la largeur de bande du signal transmis} \quad (1.6)$$

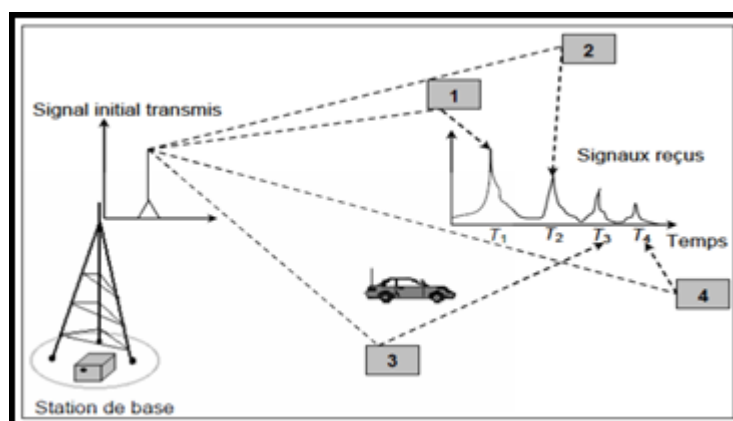


Figure 1.6: Etalement temporel

1.4.1.3.2 Effet Doppler :

La variation du canal de propagation est liée aux mouvements des différents éléments du milieu de propagation : mouvement de l'émetteur et/ou du récepteur. Les variations du canal sont proportionnelles à la vitesse de déplacement de l'objet avec lequel l'onde interagit ainsi qu'à la fréquence porteuse. Ce phénomène s'appelle l'effet Doppler. On peut considérer l'effet Doppler comme le fréquentiel de l'étalement temporel, et définir ainsi un étalement fréquentiel correspondant à la différence entre le plus grand et le plus petit décalage en fréquence inhérents aux multiples trajets [8]. On représente par le temps de cohérence du canal, durant lequel les distorsions temporelles du canal restent négligeables. Traditionnellement, est du même ordre de grandeur que l'inverse de l'étalement fréquentiel :

$$T_c = \frac{1}{B_m} \quad (1.7)$$

On note T_s la période

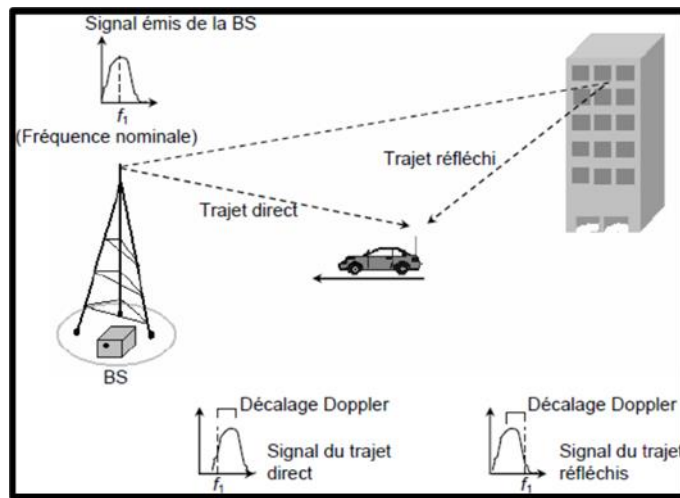


Figure 1.7: Effet doppler

1.4.1.3.3 Canal multi-trajets :

On parle de multi-trajets lorsqu'un signal émis suit une multitude de trajets de propagation depuis l'antenne jusqu'au récepteur [8]. Ces chemins forment le signal reçu. Il est ainsi composé des rayons réfléchis ou diffractés, arrivant avec un retard par rapport au rayon direct, ils sont parfois appelés des échos. La modélisation du canal multi-trajets prend en compte l'impact des différents trajets dus à la diversité de propagation dans le milieu indoor (réflexions multiples) et d'observer l'influence des autres utilisateurs dans la même bande. L'équation modélise un canal comprenant i utilisateurs et n multi-trajets

$$r(t) = \sum_{n=0}^i \sum_{i=0}^n a_n S_i(t - \tau_n) \quad (1.8)$$

Où $r(t)$ est le signal en sortie du canal, a_n et τ_n sont respectivement l'atténuation et le retard du trajet n et $S_i(t)$ est le signal émis par l'utilisateur i .

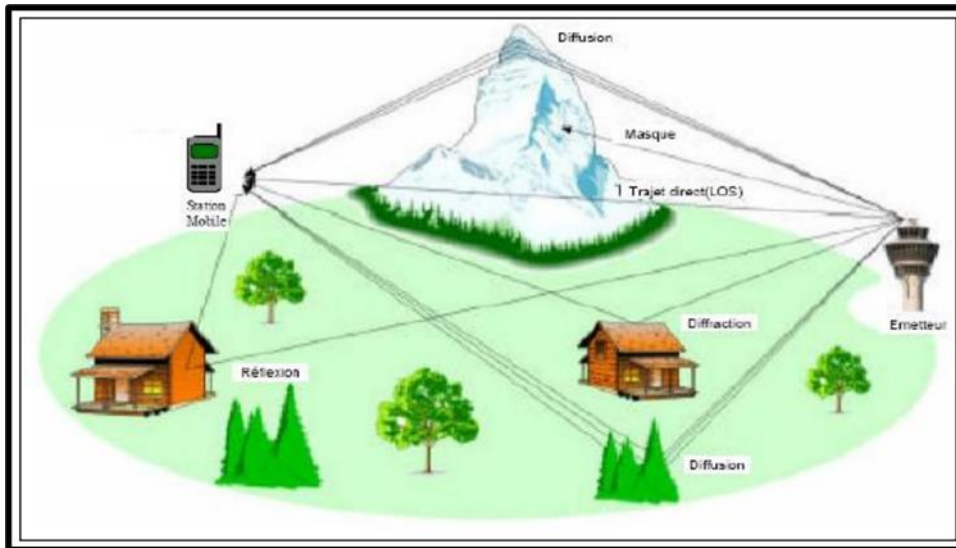


Figure 1. 8: Illustration du phénomène de trajets multiples sur le canal radio-mobile

1.4.1.3.4 Canal de Rayleigh :

C'est la distribution qui est utilisée pour modéliser les évanouissements dus aux multi-trajets incohérents lorsque l'émetteur et le récepteur ne sont pas en vue directe (NLOS : Non Line Of Sight). Il s'agit d'un des canaux de propagation les plus difficiles, mais assez commun dans les environnements urbains denses.

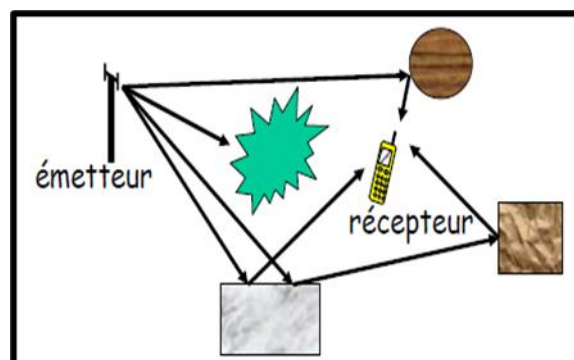


Figure 1.9: Canal de Rayleigh

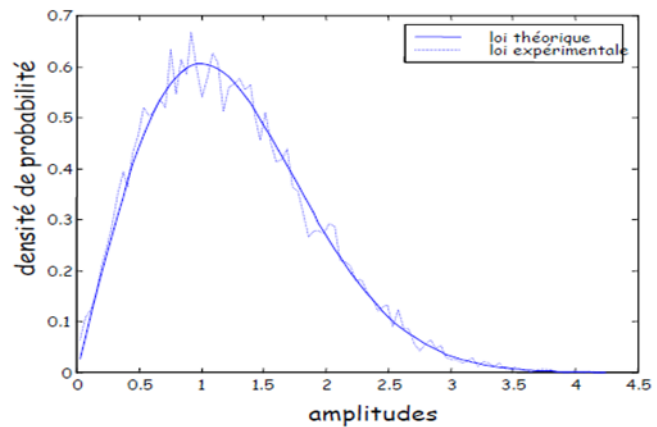
Ce modèle permet de caractériser les variations rapides de la force du signal reçu dues aux changements de phase lorsqu'un terminal mobile se déplace sur de petites distances proches de quelques longueurs d'onde ou sur des durées courtes de l'ordre de quelques secondes. Puisque la puissance moyenne reste constante sur ces petites distances, les évanouissements à petite échelle peuvent être considérés comme superposés à des évanouissements à grande échelle pour les modèles à grande échelle [8].

$$y(t) = \alpha(t)s(t) + n(t) \quad (1.9)$$

$\alpha(t)$: Variable aléatoire gaussienne complexe de moyenne **nulle**

$r(t)=|\alpha(t)|$: variable de Rayleigh, de densité de probabilité :

$$p(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad r \geq 0$$



1.4.1.3.5 Canal de Rice :

Ce modèle est souvent utilisé pour caractériser un environnement de propagation y compris une composante spéculaire forte due à une vue directe entre l'émetteur et le récepteur (LOS : Line Of Sight), et une multitude de trajets incohérents dus aux diffuseurs de l'environnement, comme la montre dans la Figure 1.10.

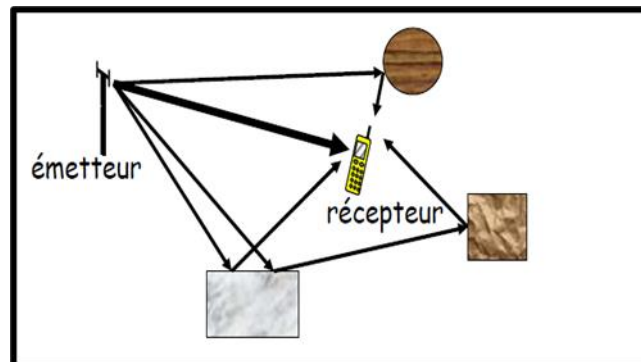


Figure 1. 10: canal de Rice

Propagation LOS : signal direct plus puissant que les autres

$$y(t) = \alpha(t)s(t) + n(t) \quad (1.10)$$

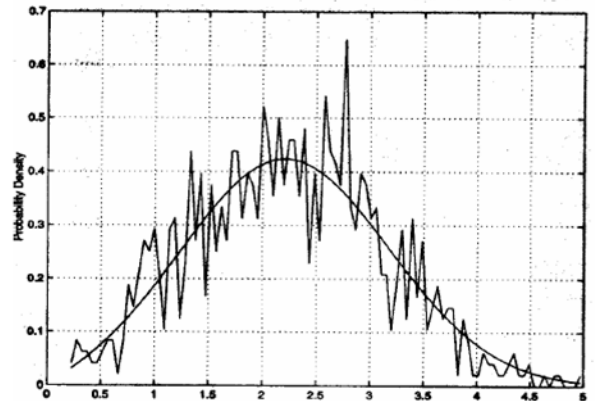
$\alpha(t)$: Variable aléatoire gaussienne complexe de moyenne **non-nulle**

$r(t)=|\alpha(t)|$: variable de Rice, de densité de probabilité :

$$p(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + s^2}{2\sigma^2}\right) I_0\left(\frac{rs}{\sigma^2}\right), \quad r \geq 0$$

s : amplitude du signal LOS

$$p(r) = \frac{r e^{-k}}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) I_0\left(\frac{r\sqrt{2k}}{\sigma^2}\right), \quad r \geq 0$$



$$k = \frac{\text{puissance du signal LOS}}{\text{puissance des signaux NLOS}} = \frac{S^2}{2\sigma^2} : \text{Paramètre de Rice} \quad (1.11)$$

1.5 Récepteur :

Le récepteur a pour objectif de reconstituer au mieux, le message émis à partir du signal reçu. Il comprend des circuits d'amplification, de changement de fréquence et de démodulation [9].

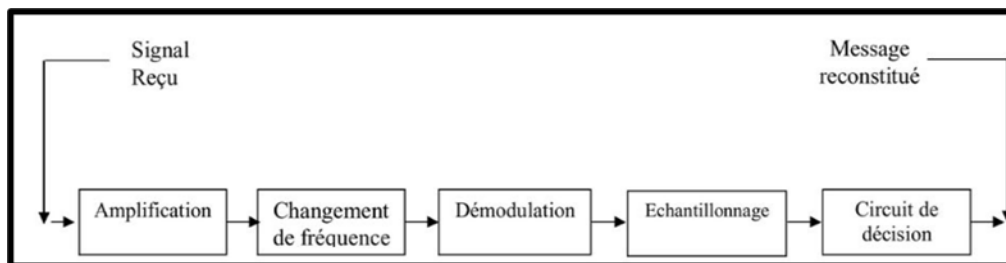


Figure 1.11: principe de récepteur pour transmission sur onde porteuse

1.5.1 Filtre de réception :

Le signal démodulé est un signal continu, mais le récepteur va devoir réaliser un échantillonnage afin de déterminer les éléments binaires transmis. Cependant, avant l'échantillonnage, on montre qu'il faut réaliser un filtrage adapté à l'émetteur pour une réception optimale des symboles transmis. Dans le cas où aucun filtre d'émission n'est employé (c'est-à-dire qu'on utilise simplement une forme d'onde, et que $g_e(t)$ est un Dirac), le filtre de réception adapté à la forme d'onde $h(t)$ a pour réponse impulsionnelle :

$$g_r(t) = h(t_0 - t) \quad (1.12)$$

Où t_0 est l'instant d'échantillonnage. Le système de réception est très simple dans ce cas, car le signal reçu à un instant donné après ce filtrage correspond directement à un symbole unique, et celui-ci peut alors être décodé. Par contre, lorsqu'un filtre d'émission est utilisé la

réponse de ce filtre est généralement plus longue que T_S , et le signal reçu à un instant t ne dépend plus d'un seul symbole émis, mais également des autres symboles [10]. Ce phénomène est appelé interférence entre symboles (IES). Pour annuler cette interférence, il faut qu'à l'instant d'échantillonnage on ne prélève que le symbole émis, et annuler l'influence due aux autres symboles. C'est-à-dire qu'il faut que la réponse impulsionnelle $r(t)$ de la chaîne de transmission complète

$$r(t) = h * g_e * g_r \quad (1.13)$$

$$r(t_0 + kT_S) = p_0 \delta_{0k} \quad \forall k \quad (1.14)$$

Où p_0 est un réel, t_0 et T_S sont respectivement l'instant et la période d'échantillonnage et δ_{0k} est le symbole de Kronecker. On dit dans ce cas que $r(t)$ vérifie le critère de Nyquist.

Le critère de Nyquist permet de déterminer l'expression du filtre de réception pour qu'il soit adapté au filtre d'émission et à la forme d'onde. De plus on peut montrer qu'il existe une répartition optimale entre les deux filtres d'émission et de réception, que l'on nomme demi-Nyquist, ou racine de Nyquist.

1.5.2 Estimateur :

En statistique, un estimateur est une fonction permettant d'évaluer un paramètre inconnu relatif à une loi de probabilité (comme son espérance ou sa variance). Il peut par exemple servir à estimer certaines caractéristiques d'une population totale à partir de données obtenues sur un échantillon comme lors d'un sondage. La définition et l'utilisation de tels estimateurs constituent la statistique inférentielle.

La qualité des estimateurs s'exprime par leur convergence, leur biais, leur efficacité et leur robustesse. Diverses méthodes permettent d'obtenir des estimateurs de qualités différentes.

1.5.3 Égalisateur :

Pour rendre l'interférence entre symboles arbitrairement faible, ou même de l'éliminer complètement à l'instant de décision, Nyquist suppose que la réponse fréquentielle du canal est constante dans la bande passante, et cela, pour répartir le filtrage entre l'émission et réception. Pratiquement, la réponse du canal n'est pas constante, et de plus, il est non stationnaire. Même si on suppose le cas idéal du canal, l'implémentation physique des filtres est approximative. Pour cela, un dispositif de correction est nécessaire à la réception. C'est le rôle de l'égaliseur.

L'objet de la fonction d'égalisation est de permettre de retrouver à partir de la séquence reçue, présentant de l'interférence entre symboles (IES) introduite par la sélectivité du canal, la séquence émise. Pour retrouver les données émises, il est donc nécessaire de concevoir un récepteur qui élimine les IES. Cette technique est appelée égalisation. L'origine du terme égalisation se comprend aisément dans le domaine des fréquences.

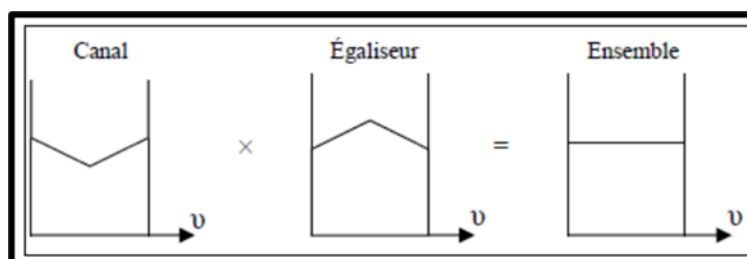


Figure 1.12: Illustration de l'égalisation spectrale.

Aspect temporel : Si la durée de réponse impulsionnelle du canal est inférieure à l'écart de temps entre deux symboles consécutifs (T_0), ceux-ci restent temporellement disjoints en sortie du canal de transmission. Dans le cas contraire, on ne peut plus les distinguer simplement les uns des autres : un traitement du signal (égalisation) s'impose. La figure (1.13) illustre schématiquement ce phénomène d'IES. La suite des symboles = $\{0, 1, 0, 1, 0, 0, 1, 0\}$ où k désigne le temps, la figure (1.13) représente l'entrée $a(t)$ du canal ainsi que sa sortie pour les deux réponses impulsionnelle H_1 et H_2 . La réponse H_1 s'étale sur exactement deux fois la durée symboles, elle induit par conséquent des interférences (il n'est plus possible de distinguer les différents symboles en sortie d'un canal de réponse H_1). La réponse H_2 quant à elle est juste inférieure à la durée maximale qui permet une transmission sans interférence : les symboles ne se recouvrent pas en sortie d'un filtre de réponse impulsionnelle H_2 [11].

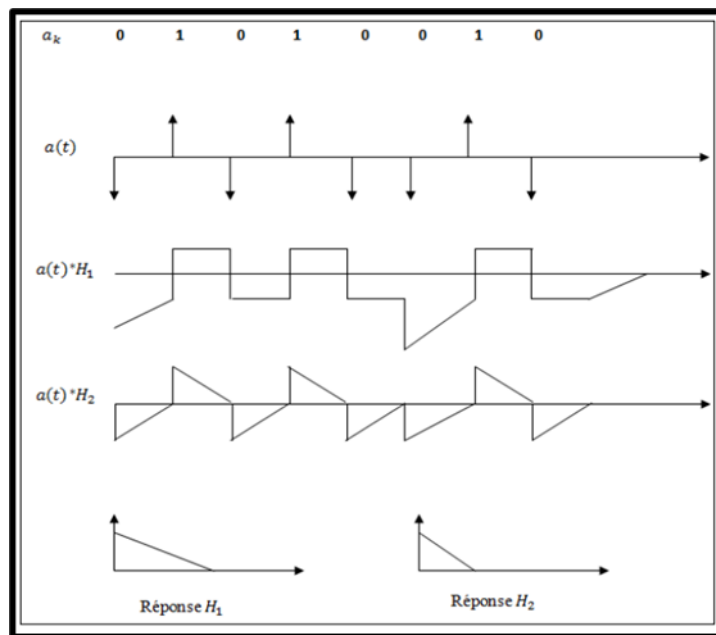


Figure 1.13: Illustration des interférences temporelles.

1.5.3.1 Démodulateur :

Le démodulateur effectue le travail inverse du modulateur, il reçoit les ondes éventuellement perturbées et les transforme en paquets de bits pour reconstituer le code sur lequel peut se trouver des erreurs de transmission [2].

Ondes $S_m(t)$ \longrightarrow démodulateur \longrightarrow mot de code

En pratique, le démodulateur doit détecter le signal entrant et fournir une mesure de fiabilité pour chaque point de la constellation. De ces mesures et de la détection dépend la probabilité d'erreur à un rapport signal à bruit donné.

1.6 Conclusion

Nous avons décrit dans ce chapitre le système de communication numérique. Nous avons détaillé une chaîne classique numérique de transmission, de l'émetteur jusqu'au le récepteur avec le canal de transmission et ses différents types. La partie émettrice comporte un mécanisme de codage utilisant un algorithme compresseur adaptatif dans le processus de codage, alors que la partie réceptrice comporte un mécanisme de codage utilisant le même algorithme synchronisé dans le processus de décodage. L'émetteur et le récepteur à chaque extrémité d'un canal de communication comportent également une interface de signalisation pouvant interrompre le processus de codage et de décodage respectivement pour émettre et recevoir une rafale de données occasionnelles sans refouler la synchronisation des états d'algorithmes compresseurs adaptatifs aux deux extrémités du canal.

CHAPITRE II : LES TECHNOLOGIES MIMO ET MASSIVE MIMO

2.1 Introduction

L'évolution de la technologie dans le domaine des télécommunications sans fil a connu ces derniers temps de nombreux changements. Plusieurs techniques, on était développé tel que la technique MIMO découverte en 1997. La technique MIMO " Multiple-Input Multiple-Output " qui signifie l'entrée multiple, sortie multiple. Cette technique de multiplexage utilisée dans les réseaux sans fil, les réseaux mobiles et les radars permettant des transferts de données à une portée plus longue et un débit plus élevé qu'avec des antennes utilisant la technique SISO (Single-Input Single-Output).

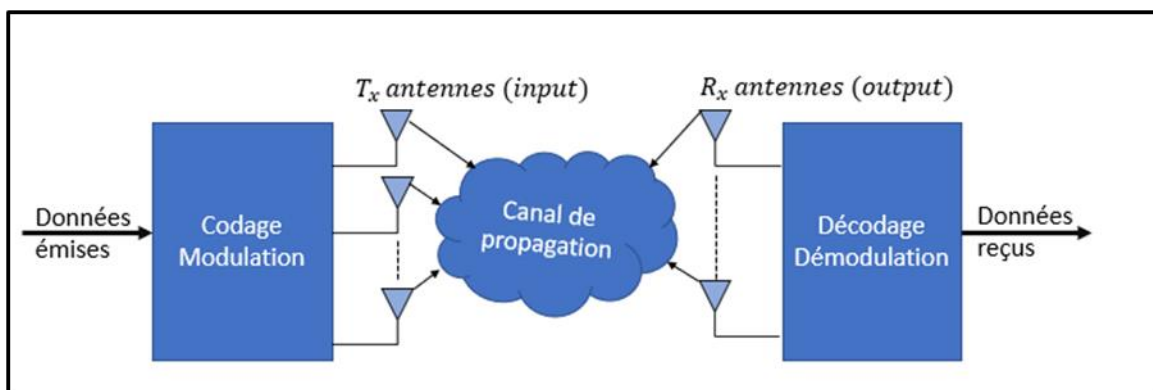


Figure 2.1: schéma général d'un système MIMO

2.2 Types de MIMO :

Selon le nombre d'antennes existant dans les différents dispositifs d'antennes, nous pouvons distinguer plusieurs configurations (SISO, SIMO, MISO, et MIMO) (systèmes). Cependant, ces architectures utilisent des différentes fonctions pour traiter le signal telles que la technique de combinaison, le codage spatio-temporel, le multiplexage spatial etc., dont le but est d'achever la meilleure performance possible du dispositif antenne [12].

2.2.1 Le système SISO (Single-Input Single-Output):

SISO est une abréviation signifiant Single Input Single Output. Un système dit SISO est donc un système possédant une seule entrée et une seule sortie. Dans ce type d'antenne, il n'y a qu'une seule émission à l'extrémité émettrice et une antenne réceptrice à l'extrémité. Cela rend SISO le plus simple à mettre en œuvre et le plus facile à concevoir parmi les quatre types d'antennes disponibles [13].

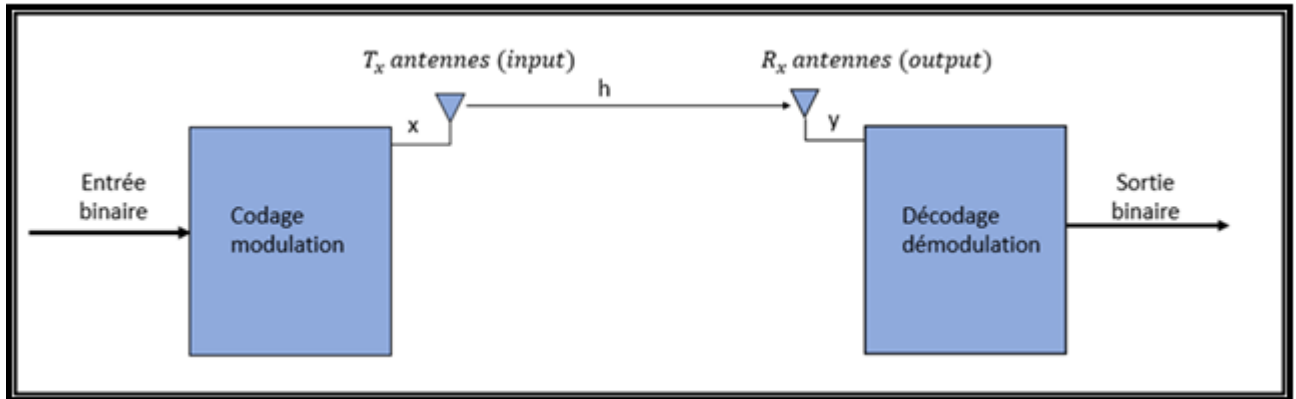


Figure 2.2: schéma du système SISO

$$\text{L'équation de ce système est : } \mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (2.1)$$

2.2.2 Le système MISO (Multiple-input Single-Output)

MISO est une Multiple-input Single-Output [14]. Dans MISO, il peut y avoir plusieurs antennes d'émission à partir desquelles le signal peut être envoyé, et il n'y a qu'une seule antenne de réception pour recevoir les signaux provenant de plusieurs antennes d'émission, ce qui signifie qu'il y a différentes sources disponibles, mais qu'il n'y a qu'une seule destination

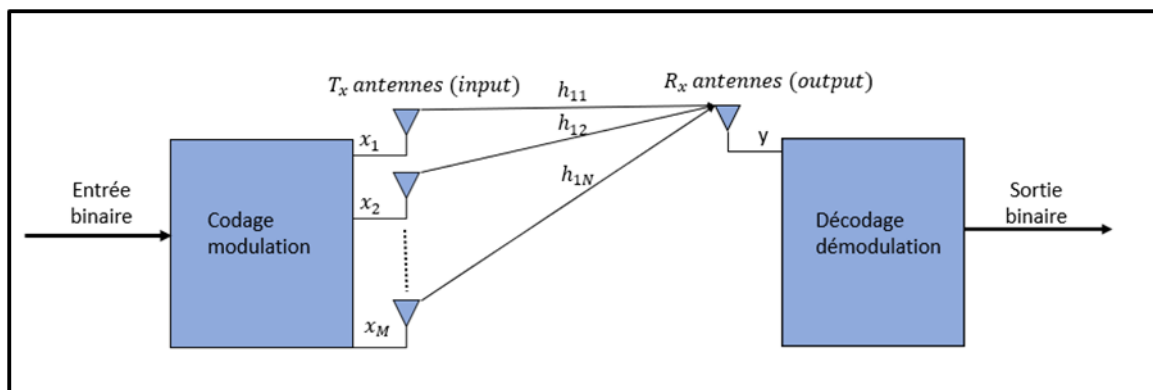


Figure 2.3 : schéma du système MISO

disponible.

$$\text{L'équation de ce système est : } \mathbf{y} = \sum_{i=1}^N H_i \mathbf{x}_i + \mathbf{n} \quad (2.2)$$

2.2.3 Le système SIMO (Single-Input Multiple-Output) :

SIMO est une abréviation signifiant Single -input Multiple-Output Un système dit SIMO est donc un système possédant une seule entrée et plusieurs sortie, cette technique a pour objectif de lutter contre l'évanouissement du canal [15].

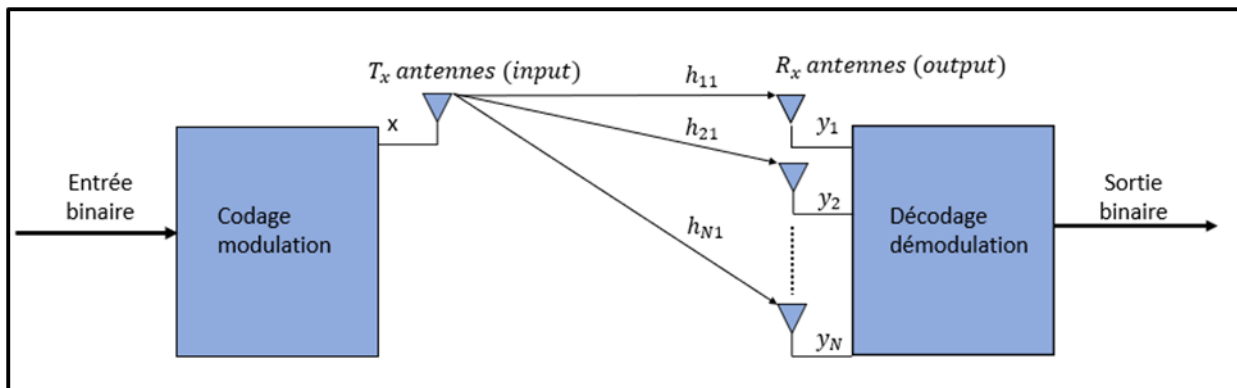


Figure 14:schéma du système SIMO

$$\text{L'équation de ce système est : } \sum_{i=1}^M y = Hx + n \quad (2.3)$$

2.2.4 Le système MIMO (Multiple-input Multiple-Output) :

MIMO est une abréviation signifiant Multiple -input Multiple-Output, c'est un système possédant plusieurs entrée et plusieurs sortie, cette configuration est la plus générale, elle englobe les systèmes MISO et SIMO comme des cas particulier. Dans MIMO, il peut y avoir plusieurs antennes d'émission à partir desquelles le signal peut être envoyé, et il existe également plusieurs antennes de réception à travers lesquelles le signal peut être reçu. Comme il peut y avoir plusieurs antennes d'émission, le signal peut être transmis par n'importe quelle antenne et donc le signal peut suivre n'importe quel chemin pour atteindre l'extrémité de réception et ce chemin suivi par le signal dépend de la position de l'antenne, c'est-à-dire si nous déplaçons l'antenne par petite position, le trajet sera modifié. L'évanouissement introduit dans le signal provenant de plusieurs trajets peut être qualifié d'évanouissement à trajets multiples.

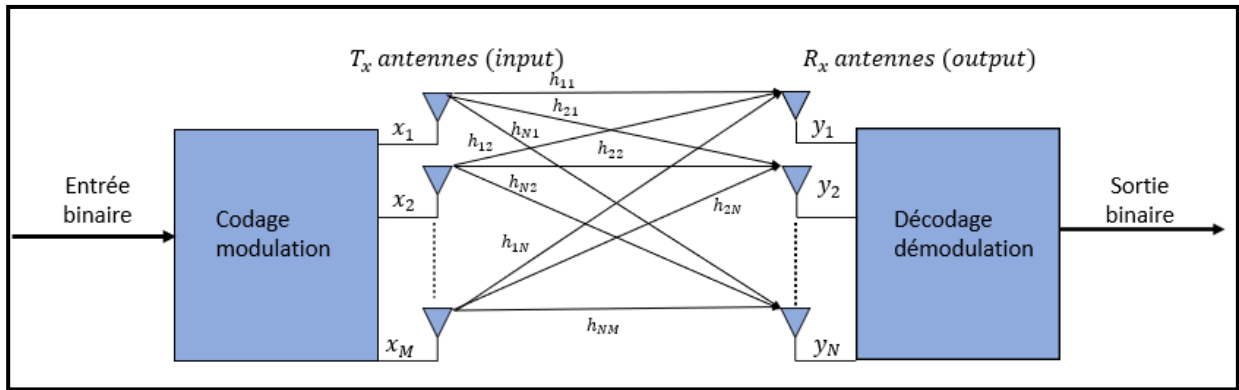


Figure 2.5: schéma du système MIMO

2.3 MODEL DE CANAL MIMO :

Un modèle de canal est nécessaire pour évaluer correctement un canal MIMO [16]. Dans MIMO, la configuration du système contient généralement M antennes à l'émetteur et N antennes à l'avant du récepteur, comme illustré dans la figure suivante :

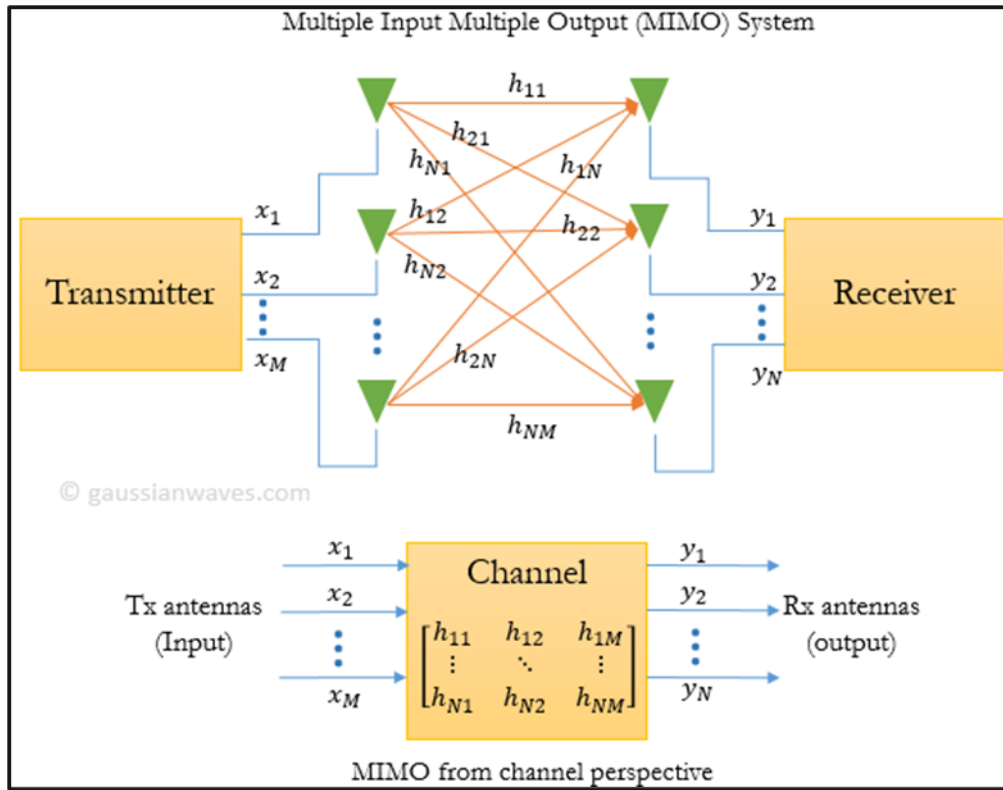


Figure 2.6: model de canal MIMO.

Ici, chaque antenne réceptrice reçoit non seulement le signal direct qui lui est destiné, mais également une fraction du signal provenant d'autres chemins de propagation. Ainsi, la réponse de canal est exprimée sous la forme d'une matrice de transmission H . Le trajet direct formé entre l'antenne 1 au niveau de l'émetteur et l'antenne 1 au niveau du récepteur est représenté par la réponse du canal h_{11} . La réponse de canal du trajet formé entre l'antenne 1 dans l'émetteur et l'antenne 2 dans le récepteur est exprimée par h_{21} et ainsi de suite. Ainsi, la matrice de canal est de dimension $N \times M$.

Le vecteur reçu (y) est exprimé en termes de la matrice de transmission de canal (H), le vecteur d'entrée (x) et le vecteur de bruit n comme :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (2.4)$$

Les différents symboles sont :

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix} \quad \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_M \end{bmatrix} \quad (2.5)$$

La réponse du lien MIMO est exprimée sous la forme d'un ensemble d'équations linéaires. Pour une configuration MIMO simple 2×2 , le vecteur de signal reçu est exprimé comme :

$$\begin{aligned} y_1 &= h_{11}x_1 + h_{12}x_2 + n_1 \\ y_2 &= h_{21}x_1 + h_{22}x_2 + n_2 \end{aligned}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (2.6)$$

Le récepteur doit résoudre cet ensemble d'équations pour découvrir ce qui a été transmis (x). La stabilité de la solution dépend du numéro de condition de la matrice de transmission H (CSI).

2.3.1 CAPACITÉ D'UN SYSTÈME MIMO :

Dans tout le canal, le signal transmis x et le bruit n sont traités avec une matrice [17]. En prenant l'allocation de puissance moyenne et l'hypothèse de matrice de canal de Rayleigh égale, la capacité du système MIMO peut être décrite avec la fonction suivante :

$$C = \sum_{i=1}^M B_i \log_2(\det [I_N + \frac{P_{t_i}}{M\sigma_n^2} HH^H]) = \sum_{i=1}^M B_i \log_2(\det [I_N + \frac{P}{M} HH^H]) \quad (2.7)$$

Où B_i , \det, I_N donnent la bande passante pour chaque utilisateur, le déterminant d'une matrice et le $M \times N$ identifie la matrice, respectivement et ρ ici est le SNR moyen dans chaque canal, H^H désigne la transposée conjuguée d'une matrice H . De plus, la matrice de canal peut être décomposée comme :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1N} \\ h_{21} & h_{22} & \dots & h_{2N} \\ \dots & \dots & \dots & \dots \\ h_{M1} & h_{M2} & \dots & h_{MN} \end{bmatrix} \quad (2.8)$$

Chacune des entrées dans la matrice de canaux de Rayleigh, comme mentionné ci-dessus, peut être exprimée comme suit:

$$h_{ij} = \alpha + j\beta = \sqrt{\alpha^2 + \beta^2} e^{j \arctan \frac{\beta}{\alpha}} = |h_{ij}| e^{j\phi_{ij}}. \quad (2.9)$$

où α et β sont des variables distribuées aléatoires et $|h_{ij}|$ est une variable distribuée aléatoire de Rayleigh. En supposant en outre que les canaux de Rayleigh sont i.i.d. les entrées de moyenne et d'unité de zéro complexe, l'expression de h_{ij} peuvent être écrites :

$$h_{i,j} = \text{Normal}\left(0, \frac{1}{\sqrt{2}}\right) + j\text{Normal}\left(0, \frac{1}{\sqrt{2}}\right) \quad (2.10)$$

où Normal donne la distribution normale. On peut donc conclure que h_{ij} obéit à la distribution χ_2^2 avec 2 degrés de liberté et $E[h_{ij}] = 1$.

Puisque l'expression de détermination de $I_N + \frac{\rho}{M} HH^H$ donne la valeur propre, on peut donc se concentrer principalement sur la valeur propre de HH^H tout en calculant la capacité. Si on rappelle que pour deux matrices $[A_{p \times q}, B_{p \times q}, p \leq q]$, la valeur propre de $AB_{p \times p}$ et

$BA_{q \times q}$ sont égales, on en conclut que les valeurs propres de HH^H et $H^H H$ sont les mêmes et que $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{\min(N,M)}]$ donne les valeurs propres non nulles (car la valeur propre nulle ne contribue pas à l'effet de capacité) du B avec :

$$\mathbf{B} = \begin{cases} HH^H, & M < N, \\ H^H H, & M > N. \end{cases} \quad (2.11)$$

Dans ce cas, la capacité MIMO peut être écrite comme :

$$C = \sum_{i=1}^M B_i \log_2 \left(\det \left[I_N + \frac{\rho}{M} HH^H \right] \right) = \sum_{i=1}^M B_i \log_2 \left(\prod_{i=1}^{\min(N,M)} \left[1 + \frac{\rho}{M} \lambda_i^2 \right] \right). \quad (2.12)$$

2.4 Les codes spatio-temporels

2.4.1 Space Time Block Coding (STBC):

Le codage par blocs spatio-temporels (STBC) est une technique utilisée dans les réseaux de communication dans le but de transmettre plusieurs copies d'un même flux de données à travers de nombreuses antennes. En conséquence, les différentes versions reçues de ces données peuvent être utilisées pour aider à améliorer la cote de fiabilité du transfert de données [18].

Alamouti a présenté le premier modèle pour le STBC en 1998. Le programme Alamouti STBC utilise deux antennes d'émission et N_r antennes de réception et peut atteindre une diversité maximale ordre de $2N_r$. Un schéma de principe du codeur spatio-temporel Alamouti est illustré à la figure suivante :

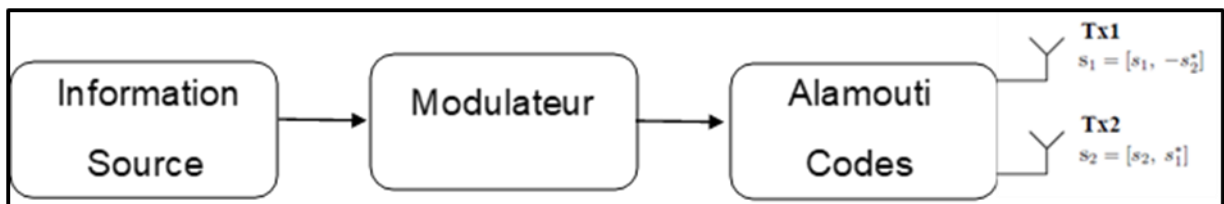


Figure 2.7 : schéma de principe du codeur spatio-temporel Alamouti

Dans le codeur se trouvent les deux symboles modulés S_1 et S_2 dans chaque opération de codage et envoyés jusqu'aux antennes d'émission sous

$$S = \begin{bmatrix} S_1 & S_2 \\ -S_2^* & S_1^* \end{bmatrix} \quad (2.13)$$

Où S_1 est envoyé depuis la première antenne et S_2 depuis la seconde antenne dans la première période de transmission. Alors que $-S_2^*$ est envoyé depuis la première antenne et S_1^* depuis la seconde antenne pendant la seconde période de transmission. Les deux lignes et colonnes de la matrice S sont orthogonales l'une à l'autre.

Le canal ressenti entre chaque antenne d'émission et de réception varie de manière aléatoire dans le temps. Cependant, le canal est supposé rester constant sur deux intervalles de temps. Les canaux h_1 et h_2 sont supposés être connus uniquement au niveau du récepteur. Un schéma de principe du décodeur spatio-temporel d'Alamouti est illustré à la figure suivante :

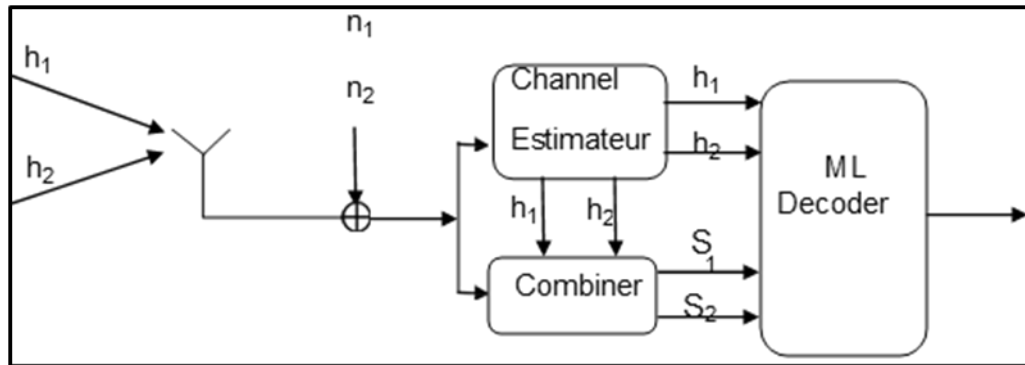


Figure 2.8: schéma de principe du décodeur spatio-temporel d'Alamouti

Dans les premiers temps, le signal reçu est :

$$y_1 = h_1 S_1 + h_2 S_2 + n_1 = [h_1 h_2] \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} + n_1 \quad (2.14)$$

Dans la deuxième tranche de temps, le signal reçu est :

$$y_2 = -h_1 S_1^* + h_2 S_2^* + n_2 = [h_1 h_2] \begin{bmatrix} -S_1^* \\ S_2^* \end{bmatrix} + n_2 \quad (2.15)$$

Où y_1 et y_2 sont le symbole reçu sur le premier et le second créneau temporel, respectivement.

n_i Est le bruit AWGN dans le $n^{ème}$ intervalle de temps, en = 1, 2.

2.4.2 Space Time Treillis Codes (STTC) :

STTC est un système de diversité de retard. Dans un tel système, le premier utilisé k bits comme antenne d'entrée pour envoyer un symbole ; la deuxième antenne le même symbole sera transmise après qu'un symbole soit retardé. Lors de la réception, pendant le décodage utilisez le maximum de vraisemblance. De ces codes tirent parti de la diversité des délais alors que sont meilleurs que Retardez la diversité [18].

- Avantages :
- Gain de codage
- Gain de diversité similaire à celui de STBC

- Désavantages :
 - Décodeur Viterbi.
 - La complexité de l'algorithme de décodage croît exponentiellement avec la longueur de la mémoire du code en treillis.

2.5 Système massive MIMO :

Le MIMO massif peut être considéré comme une forme de formation de faisceaux au sens plus général du terme, mais est assez éloigné de la forme traditionnelle. « Massive » se réfère simplement au grand nombre d'antennes dans le réseau d'antennes de la station de base. « MIMO » fait référence au fait que plusieurs utilisateurs spatialement séparés sont pris en charge par le réseau d'antennes dans la même ressource de temps et de fréquence.

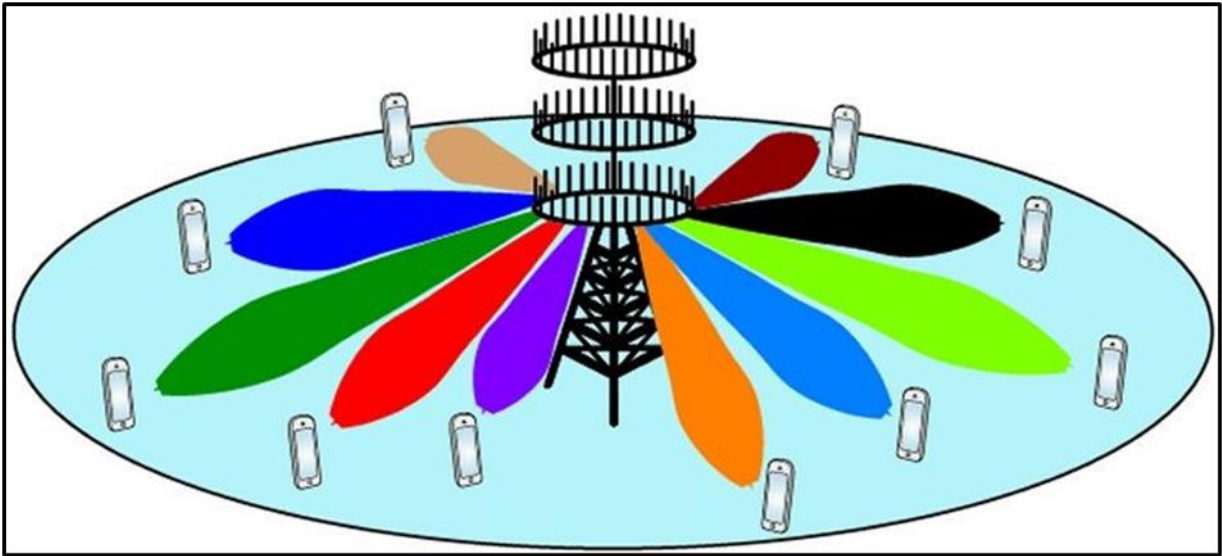


Figure 2.9 : illustration de MIMO massive et de beamforming

Ce dernier reconnaît également que dans les systèmes du monde réel, les données transmises entre une antenne et un terminal utilisateur - et vice versa - subissent un filtrage de l'environnement environnant. Le signal peut être réfléchi par les bâtiments et autres obstacles, et ces réflexions auront un retard, une atténuation et une direction d'arrivée associés (Figure : 2.10). Il se peut même qu'il n'y ait pas de ligne de visée directe entre l'antenne et le terminal utilisateur. Il s'avère que ces voies de transmission non directe peuvent être exploitées comme une puissance pour de bons.

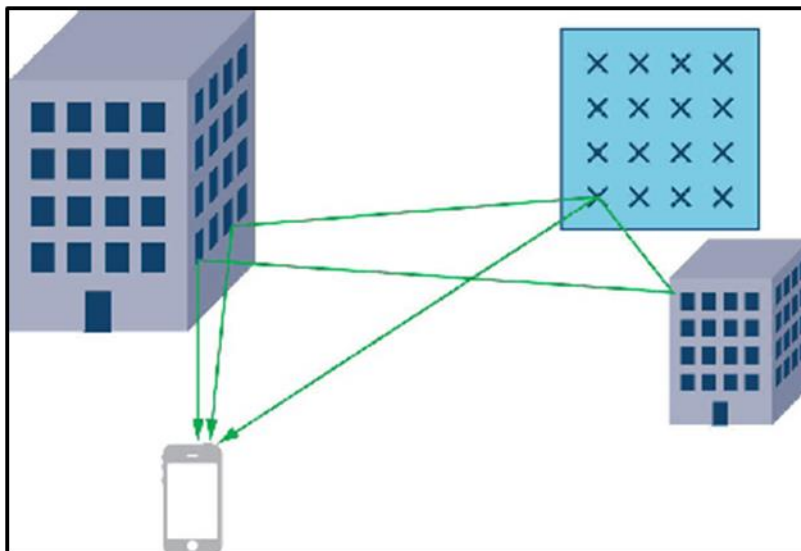


Figure 2.10: Massive MIMO

Afin de tirer parti des trajets multiples, le canal spatial entre les éléments d'antenne et les terminaux utilisateurs doit être caractérisé. Dans la littérature, cette réponse est généralement appelée information d'état de canal (CSI). Ce CSI est en fait un ensemble des fonctions de transfert spatial entre chaque antenne et chaque terminal utilisateur. Ces informations spatiales sont rassemblées dans une matrice (Figure 17.11). La section suivante examine le concept de CSI et comment il est collecté. Le CSI est utilisé pour coder et décoder numériquement les données transmises et reçues par le réseau d'antennes.

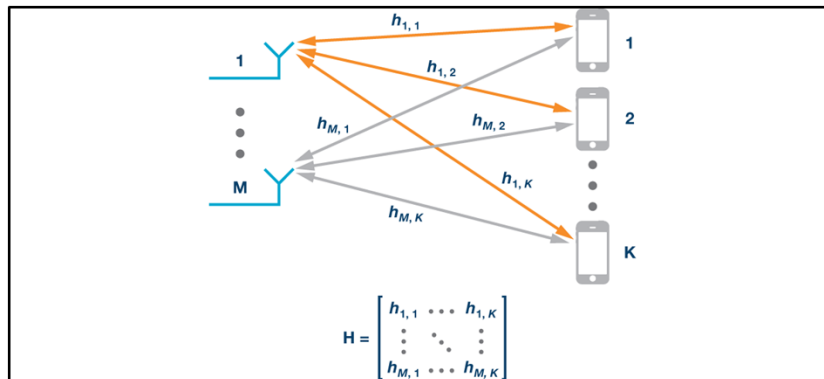


Figure 2.11: illustration de rassemblement des informations spatiales dans une matrice

2.6 Modèle de canal pour mmWave Massive MIMO :

La forte perte de trajet en espace libre est une caractéristique de la propagation mmWave, conduisant à une sélectivité spatiale limitée ou diffusion. D'autre part, les grands réseaux d'antennes étroitement compactés sont des caractéristiques des émetteurs-récepteurs mmWave, conduisant à des niveaux élevés de corrélation d'antennes. Cette caractéristique des tableaux très serrés dans des environnements à diffusion clairsemée rend de nombreuses distributions statistiques d'évanouissement utilisées dans l'analyse MIMO traditionnelle incorrecte pour la modélisation de canaux mmWave. Pour cette raison, la représentation de canal en cluster à bande étroite, basée sur le modèle étendu de Saleh-Valenzuela, qui nous permet de capturer avec précision les caractéristiques dans les mmWave channels. Lors de l'utilisation du modèle de canal de cluster, on suppose que la matrice du canal H est la somme des contributions du chemin de propagation L . Par conséquent, le canal H dans l'horloge discret peut être écrit comme :

$$H = \sqrt{\frac{N_t N_r}{L}} \sum_{l=1}^L \alpha_l \Lambda_r(\phi_l^r, \theta_l^r) \Lambda_t(\phi_l^t, \theta_l^t) a_t(\phi_l^r, \theta_l^r) a_r(\phi_l^t, \theta_l^t).$$

(2.16)

Où α_l est le gain complexe du l-ème chemin, et $\phi_l^r(\theta_l^r)$ et $\phi_l^t(\theta_l^t)$ sont les angles d'azimut (angles d'élévation) d'arrivée et de départ (AoAs / AoDs), respectivement. Les fonctions $\Lambda_r(\phi_l^r, \theta_l^r)$ et $\Lambda_t(\phi_l^t, \theta_l^t)$ représentent les gains des éléments d'antenne du récepteur et de l'émetteur sous les AoA/AoD correspondants. Pour simplifier, mais sans perte de généralité, $\Lambda_r(\phi_l^r, \theta_l^r)$ et $\Lambda_t(\phi_l^t, \theta_l^t)$ peuvent être définis comme un seul vecteur dans la gamme des AoD/AoAs (angles of departure or arrival). Enfin, les vecteurs $a_t(\phi_l^t, \theta_l^t)$ et $a_r(\phi_l^r, \theta_l^r)$ représentent respectivement les vecteurs de réponse normalisés des réseaux récepteur et émetteur à l'azimut (angle d'élévation) de $\phi_l^r(\theta_l^r)$ et $\phi_l^t(\theta_l^t)$ en fonction de la structure de l'antenne de l'émetteur et du récepteur. Pour un tableau linéaire unifié (ULA : Uniform Linear Array) avec N éléments, le vecteur de réponse du tableau peut être exprimé comme :

$$a_{ULA}(\phi) = \frac{1}{\sqrt{N}} [1, \dots e^{j\frac{2\pi}{\lambda}d \sin(\phi)}, \dots e^{j(N-1)\frac{2\pi}{\lambda}d \sin(\phi)}]_T \quad (2.17)$$

λ désigne la longueur d'onde du signal et d est l'espacement entre les éléments. Notez que nous n'incluons pas θ dans les arguments d'aULA car la réponse du tableau est invariante dans le domaine d'élévation. Dans le cas d'un réseau planaire uniforme (UPA : Uniform Planar Array) avec des éléments W_1 et W_2 ($W_1 W_2 = N$) respectivement horizontaux et verticaux, le vecteur de réponse du tableau peut être donné par :

$$a_{UPA}(\phi, \theta) = \frac{1}{\sqrt{N}} [1, \dots e^{j\frac{2\pi}{\lambda}d(x \sin(\phi) \sin(\theta) + y \cos(\theta))}, \dots e^{j\frac{2\pi}{\lambda}d((W_1-1) \sin(\phi) \sin(\theta) + (W_2-1) \cos(\theta))}]_T \quad (2.18)$$

Parmi eux, $0 \leq x \leq W_1 - 1$ et $0 \leq y \leq W_2 - 1$. Les UPA attirent l'attention dans la formation de faisceaux mmWave car ils fournissent une taille de réseau d'antennes plus petite, facilitent le regroupement de plus d'éléments d'antenne dans un réseau de taille raisonnable, et s'activent dans le domaine d'élévation Beamforming (également connu sous le nom de "Formation de faisceaux 3D).

2.7 Technique de precodage pour les systemes Massive MIMO :

2.7.1 Pré-codage numérique:

Le pré-codage numérique (Digital precoding) est une technologie largement utilisée dans les systèmes MIMO basse fréquence. L'idée de base du pré-codage numérique est de contrôler la phase et l'amplitude du signal d'origine pour éliminer les interférences à l'avance. D'une manière générale, le pré-codage numérique peut être divisé en deux catégories : le pré-codage linéaire, dans lequel le signal transmis est composé d'une combinaison linéaire de signaux originaux ; le pré-codage non -linéaire, dans lequel le signal transmis est obtenu de manière au non -linéaire. De plus, le pré-codage numérique peut également être divisé en pré-codage mono-utilisateur et pré-codage multi-utilisateur.

- Forçage à zéro (ZF) :

ZF élimine les interférences en transmettant le signal vers l'utilisateur prévu tout en annulant dans les directions d'autres utilisateurs par la matrice suivante :

$$W_{ZF} = (H^H H)^{-1} H^H \quad (2.19)$$

- Erreur Quadratique Moyenne Minimale (MMSE) :

Avec cette technique, la matrice utilisée pour annuler l'effet du canal est un peu modifiée comme suite :

$$W_{MMSE} = (H^H H + \sigma_2^2 I)^{-1} H^H \quad (2.20)$$

2.7.2 Formation de faisceau analogique:

La formation de faisceaux analogique (Analog beamforming) est en qui effectue le contrôle MIMO et beamforming au niveau RF (Radio Frequency). Dans ce cas, on peut considérer que l'émetteur-récepteur contrôle le réseau d'antennes. Le traitement du réseau d'émission et de réception est effectué par des composants RF avec des fonctions de déphasage et de réglage du gain potentiel [19].

Dans un tel système, une seule chaîne RF Comme le montre la figure (2.12) , une PLL avec de nombreux déphaseurs analogiques est utilisée pour transmettre un seul flux de données. La formation de faisceaux analogique est utilisée pour contrôler la phase signal d'origine

pour obtenir le gain maximum et le SNR effectif du réseau d'antennes. La formation de faisceaux analogique a une structure matérielle de base. Cependant, cette formation de faisceau a un faible gain d'antenne et souffre de graves pertes de performances dues uniquement à la phase d'émission. Le signal peut être contrôlé, mais son amplitude ne peut pas être contrôlée. Par conséquent, il n'est pas réellement utilisé pour Massive MIMO et Système de communication MmWave [20].

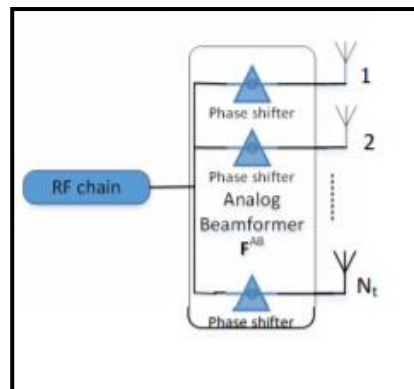


Figure 2.12: Formation de faisceau analogique .

2.7.3 Pré-codage hybride :

Le pré-codage hybride, une combinaison de pré-codage analogique et numérique, est une tentative de parvenir à un compromis entre complexité et performances. En exploitant plus d'une chaîne de radiofréquence, le pré-codage hybride permet à un système à ondes millimétriques (mmWave) de tirer parti à la fois du multiplexage spatial et du gain de formation de faisceau. Un défi majeur avec le pré-codage hybride est sa configuration dans les systèmes à large bande, car les poids de formation de faisceaux analogiques doivent être les mêmes sur toute la bande. Cette limitation est une condition difficile dans les systèmes OFDMA où plusieurs utilisateurs avec différents modèles de faisceaux préférables doivent être attribués en même temps [21].

La comparaison entre formation de faisceau analogique, Pré-codage numérique et hybride est indiqué dans le tableau suivant [22] :

Features	Beamforming types		
	Analog Beamforming	Digital Precoding	Hybride Precoding
Number of streams	Single stream	Multi-stream	Multi-stream
Number of users	Single-user	Multi-user	Multi-stream
Signal Control Cabability	Phase control only	Phase and amplitude control	Phase and amplitude control
Degree of freedom	Least	Highest	intermediate
Implementation	Phase Shifters	ADC /DAC,mixer	Phase Shifters ,ADC /DAC and mixer
Hardware Requirement	Least	Highest	Intermediate
Energy Consumption	Least	Highest	Intermediate
Cost	Least	Highest	Intermediate
Complexity	Least	Highest	Intermediate
Performance	Least	Highest	Intermediate
Suitability for mmWave Massive MIMO	Unsuitable	Impractical	Is realistic

Tableau 2.1 : tableau comparatif entre formation de faisceau analogique, Pré-codage numérique et hybride.

2.8 Simulation des systèmes de pré-codage Massive MIMO :

Sur la figure (2.13) nous considérons un système Massive MIMO avec une station de base qui emploie 64 antennes avec 4 utilisateurs, chacun ayant 16 antennes. Le pré-codage utilisé de type ZF (Forçage à zéro), $W_T = 16, N_R = 64$, canal de Salah-Venezuela.

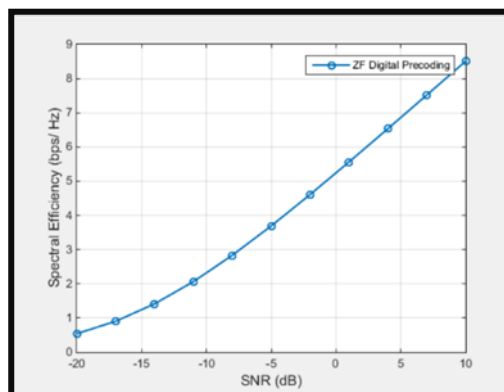


Figure 2.12: Efficacité spectrale avec ZF digital pré-coding en fonction de SNR

L'efficacité spectrale c'est le rapport de nombre de bits pouvant être transmis par unité de la bande passante, à travers l'efficacité spectrale on peut conclure la bonne ou la mauvaise utilisation de la bande passante du canal. D'après la figure on conclut que lorsqu'on augmente le nombre d'antennes de puissance (SNR), l'efficacité spectrale augmente.

Sur la figure (2.14) nous considérons un système Massive MIMO avec une station de base qui emploie 64 antennes avec 4 utilisateurs, chacun ayant 16 antennes. Les pré-codages utilisés de type MRT, ZF et MMSE, canal de Salah-Venezuela

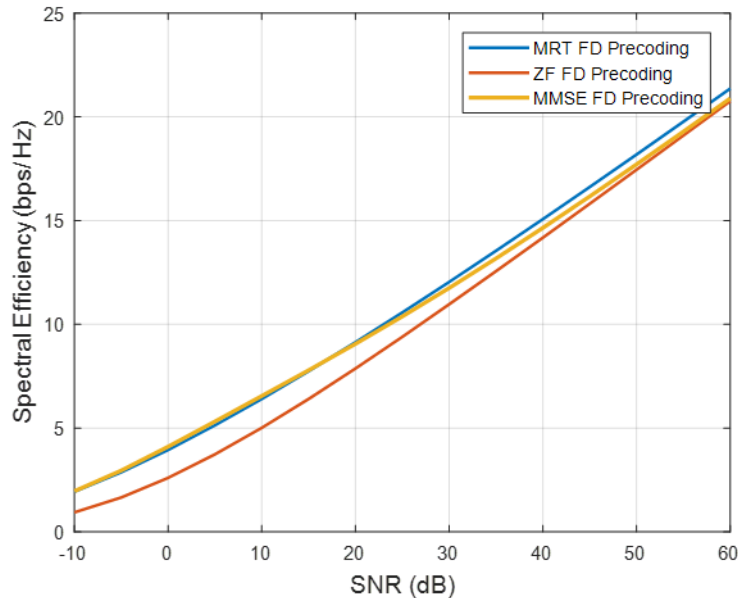


Figure 2.13: efficacité spectrale des pré-codage MRT, ZF et MMSE en fonction de SNR.

D'après la figure (2.14) on conclut que lorsqu'on augmente le nombre d'antennes de puissance (SNR), l'efficacité spectrale augmente.

2.9 Conclusion

Les systèmes MIMO massifs fournissent un débit et une efficacité spectrale beaucoup plus élevée que les systèmes MIMO traditionnels. C'est grâce à l'utilisation de nombreuses antennes. Dans ce chapitre, nous avons décrit le passage d'un système MIMO à un système Massive MIMO en donnant leurs techniques de pré-codage et la différence entre eux. La technique MIMO massive permet d'améliorer considérablement la capacité des canaux, l'efficacité énergétique et l'efficacité spectrale du système de télécommunications sans fil.

CHAPITRE III : LES SYSTEMES PROGRAMMABLES FPGA

3.1 Introduction

Ces dernières années, le développement de l'industrie a connu une évolution considérable bénéficié de la concurrence entre les fabricants. Actuellement, la logique programmable est la plus utilisée par les microcontrôleurs (μc) et microprocesseurs (μp) mais l'inconvénient de ces circuits et c'est qu'on peut seulement les programmer selon le programme existant dans une mémoire, l'architecture interne est celle proposée par le fabricant, tout comme les entrées/sorties, et aussi le nombre de ces circuits nécessaires qui peut être important, ce qui avait pour conséquence un prix important, une mise en œuvre complexe et un circuit imprimé de taille. Pour diminuer le coût de fabrication, de développement et de maintenance les PLD ont subi une évolution technologique au fil du temps depuis l'apparition de premier PAL (programmable array logique réseau logique programmable) jusqu'à l'aboutissement des premiers circuits intégrés reconfigurables du type FPGA (Field Programmable gate Arrayou réseau de cellules logiques programmable) par la société XILINX en 1985.

3.2 Les Réseaux logiques programmables PLD : (PLD : Programmable Logic Device)

3.2.1 Définitions des PLD :

Les circuits programmables, ou autrement dit les réseaux logiques programmables, sont un ensemble d'opérateurs combinés. Combinateur universel avec circuit programmable mémoire (réseau logique) ou fonction standard (multiplexeur et OU figure. (3.1)) et des interrupteurs à bascule dont les fonctions ne sont pas figées pendant le processus de fabrication. Il contient la possibilité d'exécuter une catégorie plus ou moins large de fonctions selon son architecture. La programmation du circuit consiste à définir une fonction tous ceux qui pourraient être réalisables. Comme toute implémentation logique les fonctions logiques câblées sont définies par l'interconnexion entre opérateurs combinaisons et déclencheurs, ainsi que les équations des opérateurs de combinaison. Lequel par conséquent, les dispositifs programmables du circuit impliquent des interconnexions et des opérateurs.

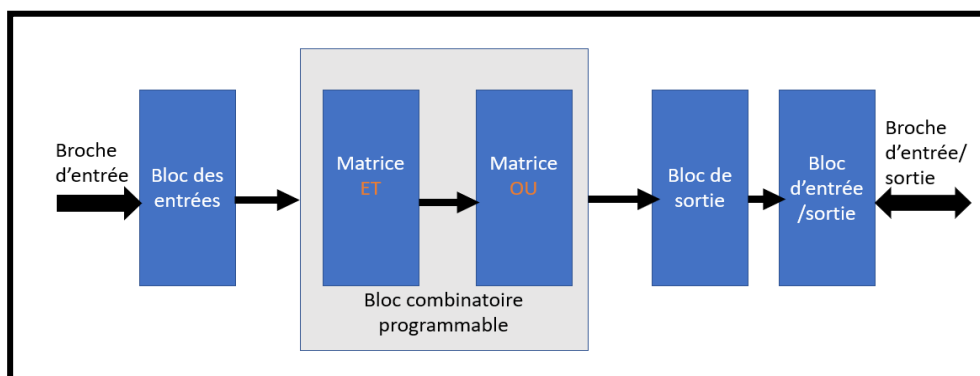


Figure 3.1: La structure de base d'un PLD

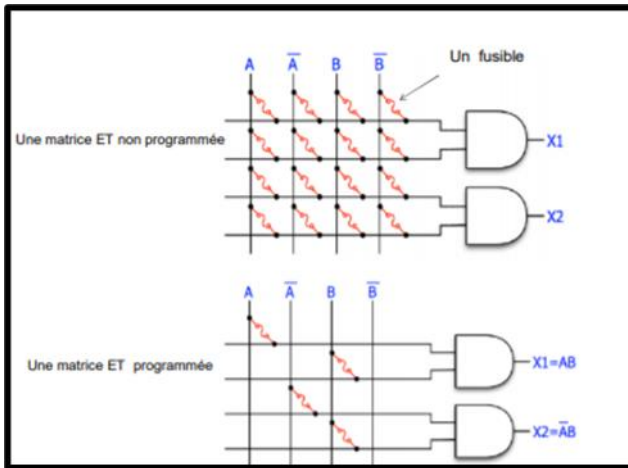


Figure 3.2: Exemple d'une matrice ET

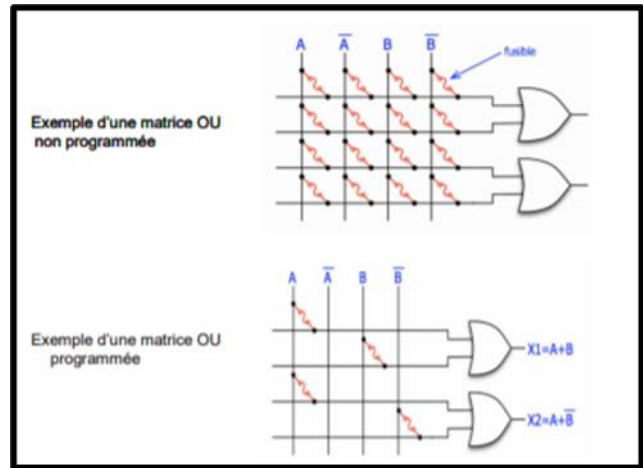


Figure 3.3: Exemple d'une matrice OU

3.2.2 Les types des PLD :

3.2.2.1 Les P.A.L. (Programmable Array Logic) :

L'idée du P.A.L est d'utiliser la technologie des fusibles. Programmation en détruisant le fusible (fusible détruit équivalent à circuit ouvert). Avec cette structure de base, il sera possible de réaliser de nombreuses fonctions logiques. La programmation constituera une destruction fusible pour obtenir la fonction requise, sachez que lors de l'achat de P.A.L tous les fusibles sont vierges ou en bon état [23].

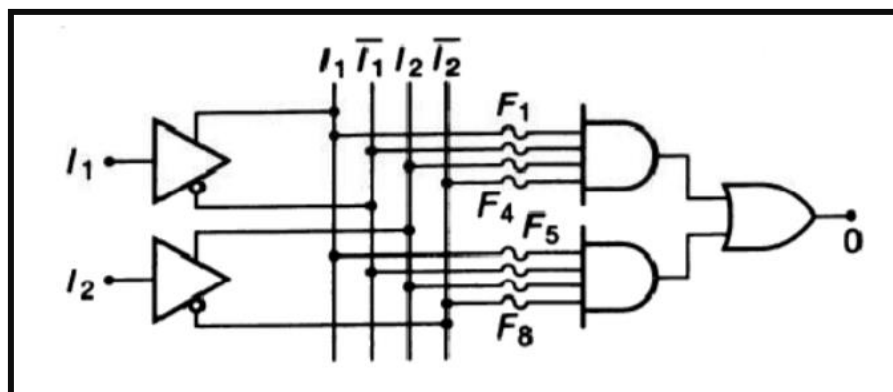


Figure 3.4: la structure simplifiée d'un PAL (2 entrées et 1 sortie)

3.2.2.2 Les G.A.L (Generic Array Logic) :

Generic Array Logic ou le réseau logique général. Titre GAL est une marque déposée de LUTTICE SEMICONDUCTEUR, c'est la première entreprise qui fournira ce type de produit sur le marché. D'autres marques proposent produits équivalents vendus sous PAL CMOS, E2PAL ou encore le nom (produits compatibles) PAL EECMOS. GAL est un PAL effaçable électriquement, utilise la Technologie CMOS.

3.2.2.3 Les EPLD : (Erasable Programmable Logic Device) :

Dispositif logique programmable effaçable, PAL effaçables. Ce sont des circuits programmables et effaçables électriquement ou par ultraviolet (UV). Ils sont encore appelés PAL CMOS. Ces circuits ont une capacité en nombre de portes et en possibilités de configuration supérieure à celle des GAL.

3.2.2.4 LES C.P.L.D (Complex Programmable Logic Device) :

Ces circuits ont le nombre et les possibilités de portes meilleures configuration que PAL. Leur architecture est basée sur ceux de PAL. Un CPLD équivaut à plusieurs PAL placés dans le même CPAL le circuit associé à la zone d'interconnexion. Le nombre de portes peut être 100 à 100 000 portes logiques [24].

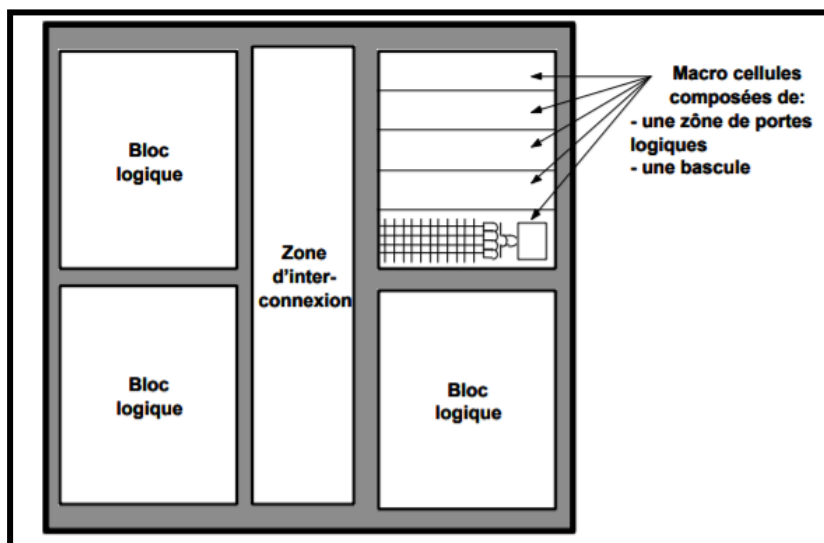


Figure 3.5: Structure générale d'un CPLD

3.2.3 **FPGA (Field Programmable Gate Array) :**

Circuit programmable composé d'un réseau de blocs logiques de base (des milliers de portes), unités d'entrée et de sortie et ressources interconnexion entièrement flexible le circuit doit être placé des outils de câblage, ses caractéristiques sont les éléments de base de l'architecture, ses techniques de programmation et ses blocs logique la puissance de ces circuits leur permet d'être des milliers, voire des millions de portes logiques et de bascules. La FPGA intègre même une mémoire vive (RAM). Les deux plus grands fabricants de FPGA sont XILINX et ALTERA [25].

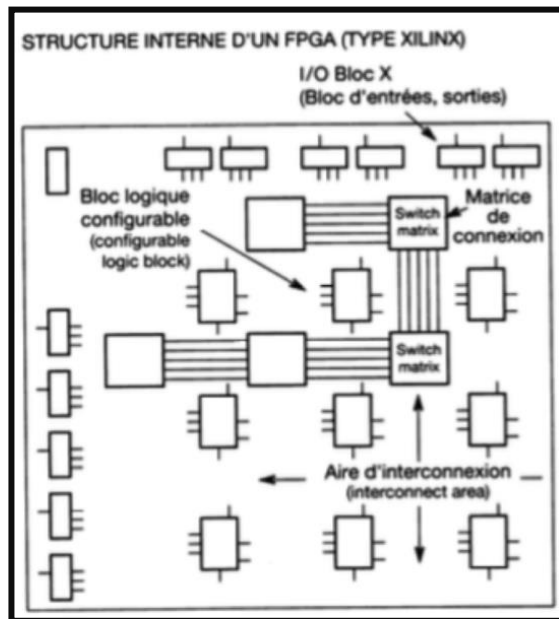


Figure 3.6: Structure interne d'un FPGA

3.3 La technologie FPGA :

3.3.1 Structure générale d'un circuit FPGA :

L'architecture interne du FPGA est différente entre les fabricants [26]. La gamme de produits du même fabricant est différente, mais rien ne peut arrêter leur similitude comme le montre la figure ci-dessous :

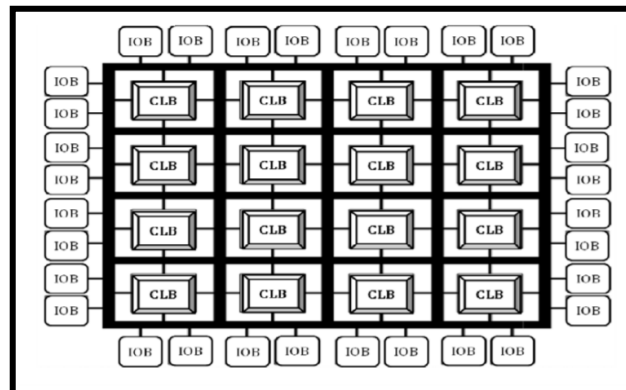
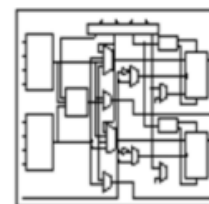


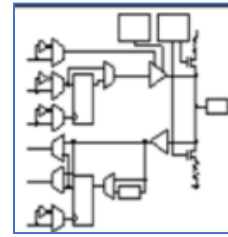
Figure 3.7: Structure général d'un circuit FPGA.

En général, un FPGA est composé des blocs principaux électriquement configurables suivants [27] :

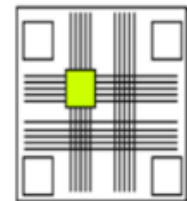
- Les blocs logiques configurables (CLB) :



- Les blocs d'entrée-sortie (IOB) :



- Les ressources d'interconnexion



3.3.2 FPGA de la famille Xilinx :

Actuellement, les circuits FPGA (bas, moyen et haut densité) sont produits par deux principaux fabricants de circuits logiques programmables : Xilinx et Altera. Il existe plusieurs autres fabricants de circuits FPGA, Exemples : Actel, Abound Logic, Achronix, Atmel, Cypress, Lattice Semi-conducteurs, etc. [28].

Le principe des FPGA de la famille XILINX est de stocker la configuration dans une mémoire vive SRAM (Static Random Access Memory) statique. Aujourd'hui des blocs de fonctionnalités supplémentaires dans quelques versions évoluées sont ajoutés et dédiés à des applications spécifiques. Ceux-ci les autres fonctions incluent : mémoire RAM (Random Access Memory), multiplicateur, bloc DSP (Digital Signal Processor), noyau de processeur RISC (Reduced Instruction Set Computer) et arbres de distribution d'horloge dans le cas d'un FPGA de grande capacité, différents domaines d'horloge peuvent réaliser une bonne synchronisation [29].

L'historique des FPGA chez Xilinx :

- FPGA (SRAM) couteux :
 - XC2000(1984)
 - XC3000(1987)
 - XC4000(1991)
 - Virtex (1999)
 - Virtex-E (2000)
 - Virtex-II (2001)
 - Virtex-II (2003)
 - Virtex-4(2005)
 - Virtex-5(2006)
 - Virtex-6(2009)

- FPGA faible coût :
 - Spartan (1998)
 - Spartan-XL (1999)
 - Spartan-2(2000)
 - Spartan-2E (2002)
 - Spartan-3(2004)
 - Spartan-6(2009)

3.4 Language VHDL :

3.4.1 Introduction :

VHDL “ very high speed integrated circuits hardware description language“, c’est un langage de haut niveau. Il permet de décrire d’un point de vue purement fonctionnel, le comportement des systèmes numériques. Cela permet également de Synthétiser les circuits numériques, c'est-à-dire convertir leurs descriptions de comportement en diagrammes logiques (nette List de ressources de base) pour programmer le composant type FPGA programmable, puis test des composants dans des systèmes complexes avant même de l’acheter ou d’attendre qu’une autre équipe le développe. L’une des particularités du VHDL provient du fait qu’il est possible d’exprimer facilement le parallélisme au sein d’un circuit.

3.4.2 Entité, Architecture et VHDL Opérateurs :

3.4.2.1 Entité :

L’entité fait partie du programme VHDL décrit les ports IN/OUT (entrées/sorties) d’un circuit électronique. Un port peut être un port d’entrée ou un port de sortie, ou les deux ports d’entrée et de sortie en même temps, ou peut être un port tampon.

Le programme VHDL pour les ports d’entrée/sortie de circuit électronique :

```

library IEEE; -- IEEE library
use IEEE.std_logic_1164.all; -- Necessary to use the std_logic

entity my_circuit_name is
  port( inp1, inp2: in std_logic; --Input ports
        outp1, outp2, outp3: out std_logic ); --Output ports
end my_circuit_name; -- End of entity

```


3.4.2.2 Les types de données utilisés dans les ports d'entrée/sortie :

Dans cette section, nous donnerons des informations sur les types de données utilisés dans VHDL Programmation.

(a) Utilisez data type 'std_logic_vector' pour décrire les ports dans la partie entité.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity FourBit_Circuit is
  port( inp1: in std_logic_vector(3 downto 0);
        inp2: in std_logic_vector(3 downto 0);
        outp1: out std_logic_vector(4 downto 0);
        outp2: out std_logic_vector(4 downto 0) );
end entity;
```

(b) Utilisez le data type 'bit_vector' pour décrire les ports dans la partie entité.

```
entity FourBit_Circuit is
  port( inp1: in bit_vector(3 downto 0);
        inp2: in bit_vector(3 downto 0);
        outp1: out bit_vector(4 downto 0);
        outp2: out bit_vector(4 downto 0) );
end entity;
```

(c) Utilisez le data type 'integer' pour décrire les ports dans la partie entité.

Nous n'avons pas besoin d'inclure d'en-tête déposer.

```
entity FourBit_Circuit is
  port( inp1: in integer range 0 to 15;
        inp2: in integer range 0 to 15;
        outp1: out integer range 0 to 31;
        outp2: out integer range 0 to 31 );
end entity;
```

Cependant, si nous supposons que le port du circuit accepte à la fois le positif et le négatif

Entiers, dans ce cas, en utilisant 4-bits, nous pouvons obtenir les entiers compris entre -8 et 7, et en utilisant 5-bits, nous pouvons obtenir des entiers compris entre -16 et 15.

Ainsi, nous pouvons écrire le segment d'entité comme suite :

```
entity FourBit_Circuit is
  port( inp1: in integer range -8 to 7;
        inp2: in integer range -8 to 7;
        outp1: out integer range -16 to 15;
        outp2: out integer range -16 to 15 );
end entity;
```

(d) Utilisez le data type 'natural' pour décrire les ports dans la partie entité.
Nous n'avons pas besoin d'inclure d'en-tête déposé.

```
entity FourBit_Circuit is
  port( inp1: in natural range 0 to 15;
        inp2: in natural range 0 to 15;
        outp1: out natural range 0 to 31;
        outp2: out natural range 0 to 31 );
end entity;
```

(e) Utilisez le data type 'positive' pour décrire les ports dans la partie entité.
Nous n'avons pas besoin d'inclure d'en-tête déposé.

```
entity FourBit_Circuit is
  port( inp1: in positive range 1 to 15;
        inp2: in positive range 1 to 15;
        outp1: out positive range 1 to 31;
        outp2: out positive range 1 to 31 );
end entity;
```

(f) Utilisez le data type 'unsigned' pour décrire les ports dans la partie entité.

```
library IEEE;
use IEEE.numeric_std.all;

entity FourBit_Circuit is
  port( inp1: in unsigned(3 downto 0);
        inp2: in unsigned(3 downto 0);
        outp1: out unsigned(4 downto 0);
        outp2: out unsigned(4 downto 0) );
end entity;
```

(g) Utilisez le data type 'signed' pour décrire les ports dans la partie entité.

```
library IEEE;
use IEEE.numeric_std.all;

entity FourBit_Circuit is
  port( inp1: in signed(3 downto 0);
        inp2: in signed(3 downto 0);
        outp1: out signed(4 downto 0);
        outp2: out signed(4 downto 0) );
end entity;
```

3.4.2.3 Architecture :

La partie architecture d'un programme VHDL décrit le comportement interne du circuit électrique. Les données reçues des ports du circuit électronique sont traitées à l'intérieur de la partie architecture et les données output de sortie sont obtenues. Le produit des données output de sortie sont envoyées aux ports de sortie du circuit électronique. La syntaxe générale de la partie architecture est représentée comme suite :

```
architecture architecture_name of entity_name is
    Declarations
begin
    Statements
end [architecture] [architecture_name];
```

3.4.2.4 VHDL Operateurs :

Les opérateurs fournis par VHDL peuvent être classés dans les catégories suivantes :

- Opérateurs d'affectation (“:=”, “<=”, “>=”).
- Opérateurs logiques :

Logical operators	and, or, nand, nor, xor, xnor, not
-------------------	------------------------------------

Shift operators	sll, srl, sla, sra, rol, ror
-----------------	------------------------------

- Opérateurs relationnels (= ; ≠ ; < ; > ; <= ; >=)
- Opérateurs arithmétiques (+ ; - ; * ; / ; ** ; mod ; rem ; abs)
- Opérateur de concaténation “&”

3.4.3 Conception de circuits logiques combinatoires et codage simultané en VHDL:

3.4.3.1 Les déclarations “When” et “Select” :

Les instructions VHDL “When” et “Select” peuvent être utilisés dans des codes VHDL simultanés. Ces instructions ne sont pas utilisées en séquentiel. Nous pouvons utiliser des instructions VHDL “When” et “Select” pour la mise en œuvre des circuits combinatoires. Ces instructions sont employées pour l'implémentation conditionnelle des circuits logiques. La syntaxe de l'instruction ‘select’ est la suivante :

```
<Signal object>      <=      <statement >  when <condition> else
                                     <statement >  when <condition> else
                                     .....
                                     <statement >  when <condition> else
                                     <statement > ;
```

3.4.3.2 Générer une déclaration :

Les instructions “When” et “Select” sont similaires. Ils ne diffèrent que par syntaxe d'implémentation. La syntaxe de l'instruction select est la suivante :

```
      when <condition> select
<Signal object>      <=      <statement >  when <condition> ,
                                     <statement >  when <condition> ,
                                     .....
                                     <statement >  whenothers ;
```

L'instruction ‘VHDL generate’ est une instruction simultanée. Il est utilisé pour générer

Plusieurs instances d'un segment de programme. L'instruction ‘generate’ a deux différentes formes qui sont ‘unconditional generate’ and ‘conditional’.

. **Génération inconditionnelle :** comme son nom l'indique n'a pas de partie conditionnelle. Sa syntaxe est comme suite:

```
Label :      for parameter in number - range generate
              [declarative_part
              Begin]
              Statements
              End generate [Label] ;
```

L'utilisation de Label est importante dans la déclaration de génération, et le mot **begin** est utilisé si la partie déclarative est disponible dans l'instruction generate.

. **Génération conditionnelle** : Sa syntaxe est comme suit :

Label : **if** condition **generate**

[declarative_part

Begin]

VHDL Statements

End generate [Label] ;

Où « condition » doit être une expression statique.

3.4.4 Types de données, tableaux et attributs :

3.4.4.1 User Defined Data Types :

En plus des types de données standard, tels qu'intégrer, std_logic, bit, etc., Nous pouvons définir nos propres types de données à utiliser dans la programmation VHDL. Les types de données définis par l'utilisateur sont décrits à l'aide du type d'instruction VHDL. La syntaxe du type est la suivante :

Type user_defined_data **is range** range_specifications ;

Les types de données définis par l'utilisateur sont généralement définis dans la partie déclarative de l'unité d'architecture ou dans l'unité de paquets.

- **Types de données, tableaux et attributs** :

Les types de données énumérés définissent par l'utilisateur sont généralement utilisés pour la mise en œuvre de machines à états finis en langage VHDL. Les types de données énumérés définissent par l'utilisateur peuvent être considéré comme un ensemble de valeurs logiques définies par l'utilisateur. La syntaxe des types de données énumérés définissent par l'utilisateur est la suivante :

Type user_defined_data **is** (value1 ; value2 ; . . . ; valueN) ;

3.4.4.2 Types de données de tableau définis par l'utilisateur :

Nous avons déjà vu les types de données, std_logic_vector, bit_vector, et ces vecteurs sont des tableaux de types de données std_logic et bit. Nous pouvons définir les vecteurs ou tableaux pour les autres types de données standards ou pour les types de données définis par l'utilisateur. Les tableaux peuvent être définis selon deux méthodes différentes.

A. Constrained Arrays :

La syntaxe des types de données de tableau contraint définis par l'utilisateur est la suivante :

Type data_array **is array** (definite array range) **of** data _type;

B. Unconstrained Arrays :

Dans les définitions de tableau sans contrainte, la taille du tableau n'est pas une valeur déterministe, c'est-à-dire une longueur indéfinie. La syntaxe de la définition de tableau sans contrainte est la suivante :

Type data_array **is array** (natural range < >) **of** data _type;

Où “Natural range < >” implique que les limites de plage du tableau doivent être comprises dans la plage de nombres naturels. Nous pouvons également utiliser une déclaration positive ou entière au lieu de natural dans la déclaration de tableau, c'est-à-dire que nous pouvons également avoir

Type data_array is array (positive range < >) of data_ type;

3.4.4.3 Définitions des tableaux 2D ou des matrices:

Les tableaux 2D peuvent être considérés comme des matrices. On peut considérer une matrice comme une colonne vecteur dont les éléments ne sont que des vecteurs de lignes, ou nous pouvons considérer une matrice comme une table de nombres et les éléments de la table sont accessibles à l'aide des index de ligne et de colonne correspondantes.

- **3D Arrays :**

Un vecteur de tableaux 2 D, c'est-à-dire un tableau de tableaux 2D, peut être considéré comme des tableaux 3D. En suivant la même logique, nous pouvons également définir des tableaux 4D, 5D,

Les tableaux 3D peuvent être définis avec deux approches différentes. Dans la première approche, nous définissons le tableau 3D via un vecteur de matrices. Dans la seconde approche, nous définissons les tableaux 3D comme une table d'entiers en 3 dimensions.

3.4.5 Subtypes :

Une déclaration de sous-type n'introduit pas de nouveau type. Bien que les opérations ne soient pas autorisées entre différents types de données, elles sont autorisées entre un type et son sous-type. Les déclarations de données de sous-types sont généralement effectuées dans la partie déclarative de l'architecture ou dans l'unité de package. La syntaxe de la déclaration de sous-type est la suivante :

subtype subtype_name **is** base_type **range** range description ;

Le type d'un sous-type est du même type que son type de base. Lorsque des objets d'un sous-type et de son type de base sont attribués l'un à l'autre, aucune conversion de type n'est nécessaire.

3.4.5.1 Type Conversion :

Une chaîne binaire peut représenter, un std_logic_vector, un vecteur de bits non signé, un vecteur de bits, un entier, une donnée naturelle ou positive. Il est possible de convertir une chaîne binaire, ou vecteur de bits, d'un type de données à l'autre via une opération de transtypage.

3.4.6 Séquentiel Circuit Implémentation in VHDL :

3.4.6.1 Séquentiel Circuits in VHDL :

Les circuits logiques numériques peuvent être divisés en deux catégories principales. Le premier est le circuit de logique combinatoire, le second est la logique séquentielle synchrone circuits, c'est-à-dire des circuits cadencés.

Dans le codage simultané, nous pouvons utiliser les instructions VHDL lorsque, **select...With, generate.**

D'autre part, dans le codage séquentiel, les instructions if, **wait**, **loop** et cas sont employés. La mise en œuvre des circuits de logique combinatoire se fait dans l'architecture unité du programme VHDL, et les circuits combinatoires sont implémentés en utilisant les codes VHDL simultanés.

D'autre part, des circuits logiques séquentiels sont mis en œuvre en utilisant des unités de programme séquentielles. Ces unités séquentielles sont le processus, la fonction, et procédure.

✓ Process :

La syntaxe de l'unité de **Process** est celle illustrée dans le programme suivant

```

process(sensitivity list)
  declaration part of the process
begin
  body part of the process
end process;

```

Un processus est utilisé pour implémenter un circuit logique séquentiel. Chaque fois qu'un paramètre dans la liste de sensibilité des changements de processus, le processus est exécuté. La variable si nécessaire, un objet peut être utilisé dans la déclaration et dans le corps du processus. La déclaration des objets de signalisation n'est pas autorisée à l'intérieur des processus.

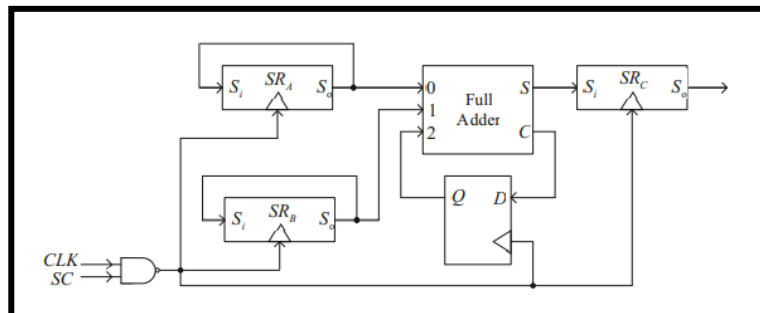


Figure 3.8: processus d'un circuit logique séquentiel

3.4.7 Packages, Components, Functions et Procédures:

3.4.7.1 Packages :

Les packages se composent de deux sections, l'une est la partie déclaration du package, l'autre est le corps du package. La partie déclaration contient généralement les déclarations de données et les en-têtes de fonction ou de procédure, c'est-à-dire les déclarations. La partie corps du package contient généralement la partie implémentation des fonctions ou procédures. La syntaxe du package est présentée dans le suivant. Pour utiliser le package dans notre programme, nous devons inclure le fichier package dans l'en-tête du programme principal.

```
package package_name is  
    declarative part  
end package_name;  
  
package body package_name is  
    body part  
end package_name;
```

Ceci est réalisé en utilisant la déclaration :

```
use work .package_name.all
```

3.4.7.2 Components:

Si certaines des unités logiques, telles que les multiplexeurs, les additionneurs, les bascules, etc., sont utilisées à plusieurs reprises dans des circuits logiques, nous pouvons écrire une bibliothèque pour ces unités et les utiliser chaque fois qu'elles sont nécessaires sans les réimplémenter encore et encore.

Ceci est réalisé via l'utilisation de l'utilitaire de composants.

Si nous avons déjà implémenté certaines des unités logiques et que nous souhaitons utiliser ces unités dans d'autres programmes VHDL, nous pouvons déclarer ces unités en tant que composants. Les déclarations des composants sont soit effectuées dans l'unité de package, soit dans la partie déclarative de l'architecture. Une fois les composants déclarés, nous pouvons les instancier en n'importe quel nombre dans notre programme principal.

La déclaration de composant est identique à la déclaration d'entité, sauf pour l'utilisation du mot-clé de composant au lieu du mot-clé d'entité. La syntaxe de déclaration et d'instanciation des composants est comme dans le programme suivant :

```
component component_name is  
    port( port_declarations);  
end component;  
  
compLabel: component_name port map(port  
list);
```

3.4.7.3 Fonctions et Procédures :

Les fonctions et procédures sont utilisées pour écrire des segments de programme séquentiels. Les fonctions et procédures sont écrites pour la mise en œuvre d'algorithmes ou de circuits fréquemment utilisés et portables. Les fonctions et procédures sont similaires à celles des processus. Cependant, les processus sont généralement employés à l'intérieur de l'unité d'architecture, par contre, les fonctions et procédures sont généralement placées dans l'unité de package pour une utilisation générale.

3.4.7.4 Fonctions :

Les fonctions peuvent être placées dans l'unité de package, ou elles peuvent être écrites dans la partie déclarative de l'unité d'architecture. Si des fonctions doivent être placées dans l'unité de package, alors une partie déclarative et une partie corps de la fonction doivent être

écrites et placées séparément dans la partie déclarative et la partie corps de l'unité de package.

La syntaxe de la fonction à l'intérieur d'un paquet est indiquée dans le programme suivant :

```
package my_package is
function function_name (input_parameter_list) return data_type;
end my_package;
-----
package body my_package is
function function_name (input_parameter_list) return data_type is
variable, constant declarations
begin
statements
return return_data;
end function_name;
end my_package;
```

3.4.7.4.1 Opérateur Overloading :

Pour augmenter la variabilité des arguments des opérateurs, nous utilisons l'approche de surcharge des opérateurs. Pour cela, nous écrivons une fonction et le nom de la fonction est choisi comme « opérateur » où l'opérateur peut être + ; - ; * ; and / etc.

La syntaxe de la surcharge de l'opérateur est représentée dans le programme suivant :

```
function "operator" (parameters) return data_type is
declarations
begin
sequential statements
end function;
```

3.4.7.5 Procedures :

Les procédures sont des unités de programme similaires aux fonctions et elles sont utilisées pour écrire des segments de programme séquentiels. Les fonctions renvoient un seul paramètre, par contre, les procédures peuvent renvoyer plusieurs paramètres. De plus, dans la liste des paramètres des procédures, nous pouvons utiliser des variables, des signaux et des constantes.

Les procédures peuvent être écrites dans la partie déclarative de l'unité d'architecture, ou elles peuvent être déclarées dans la partie déclarative du package, et leur corps peut être écrit dans la partie corps de l'unité de package. La syntaxe de l'unité de procédure est montrée dans le programme suivant :

```
procedure procedure_name(input and output parameters) is
declarations
begin
sequential statements
end procedure;
```

Pour définir les nombres à virgule fixe signés fixes et non signés, nous utilisons la syntaxe

Signal/variable snum: **sfixed** (a **down to** b);

Signal/variable unum: **ufixed** (a **down to** b);

Où $a + 1$ et $|b|$ indique le nombre de bits utilisés pour la partie entière et fractionnaire des nombres si $a > 0$ et $b < 0$. Les nombres signés sont représentés sous forme de complément à 2's.

3.4.7.6 Nombres a virgule flottante:

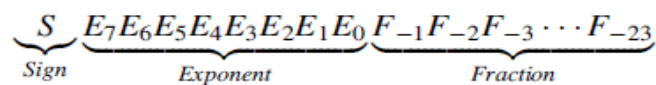
Les nombres à virgule flottante permettent une implémentation de haute précision. Cependant, tout matériel construit à l'aide des nombres à virgule flottante est au moins trois fois plus complexe que le matériel construit à l'aide des nombres à virgule fixe mettant en œuvre le même algorithme ou circuit. De plus, les circuits conçus en utilisant les nombres à virgule flottante sont beaucoup plus lents que les circuits construits en utilisant les nombres à virgule fixe. Compte tenu de ces facteurs, même si au début, les nombres à virgule flottante peuvent être très attractifs, cependant, du point de vue de la mise en œuvre, ils ne sont pas trop souhaitables.

- **Nombres a virgule flottante:**

Pour définir un nombre à virgule flottante fixe ou non signé, nous utilisons la syntaxe

Signal /variable num: **float** (a **down to** b)

Le format du nombre à virgule flottante 32 bits est comme dans



Où S indique le bit de signe, E_i est utilisé pour l'exposant et F_i est utilisé pour la partie fractionnaire.

La valeur du nombre à virgule flottante est calculée à l'aide de

$$\text{Value} = (-2S + 1)(2^{\text{Exponent Value} - \text{Exponent base}})(1 + \text{Fraction value})$$

Où la base de l'exposant est calculée comme

$$\text{Exponent base} = 2^{|E|-1} - 1$$

Où $|E|$ indique le nombre de bits utilisés dans la partie exposant. La plus petite valeur de la largeur de l'exposant et de la largeur de la fraction est 3, c'est-à-dire flottante (3 jusqu'à -3), le plus petit nombre à virgule flottante est un nombre de 7 bits.

3.5 Environnement ISE :

L'environnement ISE est un logiciel de programmation de la famille XILINX. Il est défini comme étant un environnement intégré de développement de systèmes numériques ayant pour objectif une implantation matérielle sur FPGA de la famille XILINX.

ISE intègre différents outils permettant de passer à travers tout le flot de conception d'un Système numérique. En effet il dispose :

- D'éditeur de textes, de schémas et de diagramme d'états.
- D'un compilateur VHDL.
- D'un outil de simulation.
- D'outils pour la gestion des contraintes temporelles.
- D'outils pour la synthèse.
- D'outils pour la vérification.
- D'outils pour l'implantation sur FPGA.

3.6 Conclusion :

Dans cette partie, nous avons vu le développement de circuits logiques programmables, ce qui a conduit à l'émergence des FPGA. Tout d'abord, nous avons introduit l'émergence des circuits programmables (PLD), puis nous avons étudié en détail les FPGA, en particulier la série Xilinx, ce qui nous a amenés à conclure que la technique FPGA est la priorité absolue des composants logiques et du développement technologique. La demande croissante de composants plus efficaces, plus économiques et à moindre coût est la principale direction de développement du FPGA. FPGA ouvre de larges perspectives pour le contrôle en temps réel. La réalisation du système de contrôle en temps réel nécessite également une bonne maîtrise des outils de modélisation et de simulation, et une bonne maîtrise du calcul en temps réel dans la phase de mise en œuvre.

**CHAPITRE IV : IMPLANTATION D'UN SYSTEME
MASSIVE MIMO SOUS CARTE FPGA**

4.1 Introduction

Le VHDL apparaît comme un moyen de décrire les circuits intégrés. La crise du cycle de vie du matériel : les circuits intégrés se compliquaient chaque jour, et le coût de leur remplacement augmentait. VHDL est né comme moyen standard de décrire les circuits. Le langage HDL comme le VHDL ou le Verilog sont utilisés pour décrire les fonctionnalités qui seront implémentées sur le composant. La description matérielle est traduite dans un fichier de configuration pour le FPGA cible.

4.2 Description de la carte FPGA xilinx spartan S3 :

La génération de FPGA Spartan-3 comprend la famille étendue Spartan-3A (plateformes DSP Spartan-3A, Spartan-3AN et Spartan-3A), ainsi que les familles Spartan-3 et Spartan-3E antérieures. Ces derniers ont des portes programmables sur site (FPGA) sont spécialement conçues pour répondre aux besoins des applications électroniques à volume élevé, telles que les produits grand public. La génération Spartan-3 comprend 25 appareils offrant des densités allant de 50 000 à 5 millions de portes logique.

La plate-forme Spartan-3 était le premier FPGA 90 nm de l'industrie, offrant plus de fonctionnalités et de bande passante par dollar qu'auparavant, établissant de nouvelles normes dans l'industrie de la logique programmable. La plate-forme Spartan-3E s'appuie sur le succès de la plate-forme Spartan-3 précédente en ajoutant de nouvelles fonctionnalités qui améliorent les performances du système et réduisent le coût de configuration. La famille Extended Spartan-3A s'appuie sur le succès de la plate-forme Spartan-3E précédente en améliorant encore la configuration et en réduisant la puissance pour fournir le coût total le plus bas. La plate-forme Spartan-3AN offre les avantages supplémentaires de la non-volatilité et de grandes quantités de flash utilisateur embarqué. La plate-forme DSP Spartan-3A étend la plage de densité et ajoute des ressources souvent requises dans les applications de traitement du signal numérique (DSP).

En raison de leur coût exceptionnellement bas, les FPGA de génération Spartan-3 sont parfaitement adaptés à une large gamme d'applications électroniques grand public, notamment l'accès à large bande, les réseaux domestiques, l'affichage/la projection et les équipements de télévision numérique.

Les FPGA de génération Spartan-3 offrent une alternative supérieure aux ASIC programmés par masque. Les FPGA évitent le coût initial élevé, les longs cycles de développement et la rigidité inhérente des ASIC conventionnels. De plus, la programmation du FPGA permet des mises à niveau de conception sur le terrain sans qu'aucun remplacement matériel ne soit nécessaire, une impossibilité avec les ASIC.

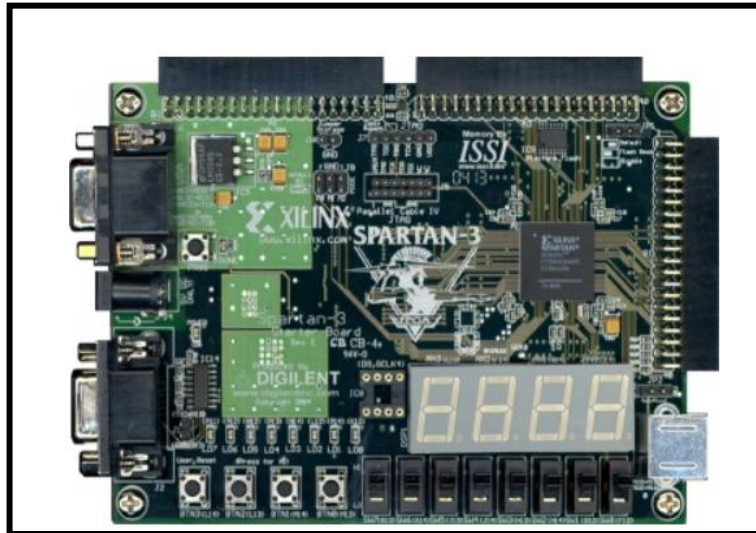


Figure 4.1: carte XILINX Spartan -3

➤ **Caractéristiques :**

- Solution logique à faible coût et hautes performances [30]
- Densités jusqu'à 74 880 cellules logiques
- Signalisation SelectIO™
- Jusqu'à 784 broches d'E/S
- Taux de transfert de données de 622 Mb/s par E/S
- 18 normes de signaux asymétriques
- 8 normes d'E/S différentielles dont LVDS, RSDS
- Terminaison par impédance à commande numérique
- Balançoire du signal allant de 1,14V à 3,45V
- Prise en charge du double débit de données (DDR)
- Prise en charge de la DDR, DDR2 SDRAM jusqu'à 333 Mbps
- Ressources logiques
- Cellules logiques abondantes avec capacité de registre à décalage
- Multiplexeurs larges et rapides
- Logique de report d'anticipation rapide
- Multiplicateurs dédiés 18 x 18
- Logique JTAG compatible avec IEEE 1149.1/1532
- Mémoire hiérarchique SelectRAM™
- Jusqu'à 1 872 Kbits de RAM de bloc totale
- Jusqu'à 520 Kbits de RAM distribuée totale
- Gestionnaire d'horloge numérique (jusqu'à quatre DCM)
- Élimination du décalage d'horloge

- Synthèse de fréquence
- Déphasage haute résolution
- Huit lignes d'horloge mondiales et routage abondant
- Entièrement pris en charge par le système de développement Xilinx ISE
- Synthèse, mapping, placement et routage
- Processeur MicroBlaze™, PCI et autres cœurs
- Options d'emballage sans plomb
- Variantes de la famille Spartan-3L à faible puissance et de la famille Spartan-3 XA pour l'automobile

➤ **L'architecture de la famille Spartan-3 :**

L'architecture de la famille Spartan-3 se compose de cinq éléments fonctionnels programmables fondamentaux [31] :

- Les blocs logiques configurables (CLB) contiennent des tables de consultation basées sur la RAM (LUT) pour implémenter des éléments logiques et de stockage qui peuvent être utilisés comme bascules ou verrous. Les CLB peuvent être programmés pour exécuter une grande variété de fonctions logiques ainsi que pour stocker des données.
- Les blocs d'entrée/sortie (IOB) contrôlent le flux de données entre les broches d'E/S et la logique interne de l'appareil. Chaque IOB prend en charge le flux de données bidirectionnel plus un fonctionnement à 3 états. Vingt-quatre normes de signaux différentes, dont sept normes différentielles hautes performances. Les registres Double Data-Rate (DDR) sont inclus. La fonction d'impédance à commande numérique (DCI) fournit des terminaisons automatiques sur puce, simplifiant la conception des cartes.
- La RAM de bloc fournit un stockage de données sous la forme de blocs à double port de 18 Kbits.
- Les blocs multiplicateurs acceptent deux nombres binaires de 18 bits comme entrées et calculent le produit.
- Les blocs Digital Clock Manager (DCM) fournissent des solutions entièrement numériques à étalonnage automatique pour la distribution, le retard, la multiplication, la division et le déphasage des signaux d'horloge.

Ces éléments sont organisés comme le montre la figure (4.2). Un anneau d'IOB entoure un réseau régulier de CLB. Le XC3S50 possède une seule colonne de blocs de RAM intégrée dans la matrice. Les appareils allant du XC3S200 au XC3S2000 ont deux colonnes de bloc RAM. Les appareils XC3S4000 et XC3S5000 ont quatre colonnes de RAM. Chaque colonne est constituée de plusieurs blocs RAM de 18 Kbits ; chaque bloc est associé à un multiplicateur dédié. Les DCM sont positionnés aux extrémités des colonnes RAM du bloc externe. La famille Spartan-3 dispose d'un riche réseau de traces et de commutateurs qui interconnectent les cinq éléments fonctionnels, transmettant des signaux entre eux. Chaque élément fonctionnel a une matrice de commutation associée qui permet des connexions multiples au routage.

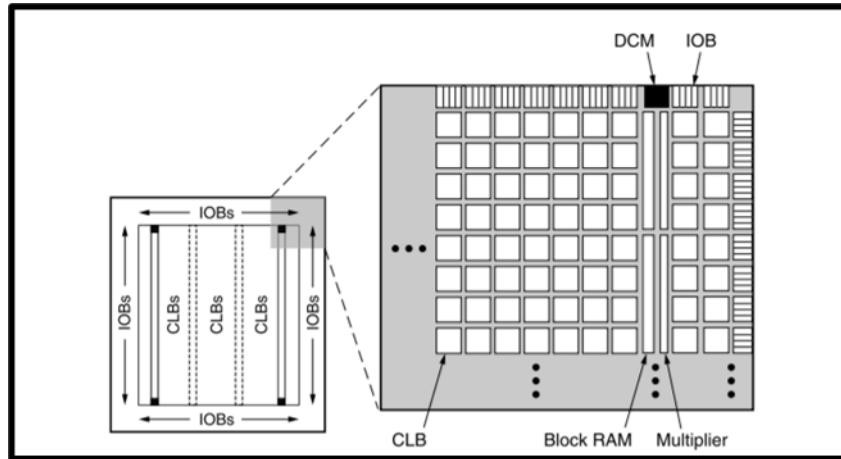


Figure 4.2: Spartan-3 Family Architecture

4.3 Descriptions du logiciel ISE xilinx : ISE (Integrated Software Environment)

Xilinx ISE (Integrated Synthesis Environment) est un outil logiciel de Xilinx pour la synthèse et l'analyse de conceptions HDL, qui cible principalement le développement de micrologiciels embarqués pour les familles de produits de circuits intégrés (IC) Xilinx FPGA et CPLD. L'utilisation de la dernière édition publiée à partir d'octobre 2013 se poursuit pour la programmation dans le système des conceptions matérielles héritées contenant des FPGA et des CPLD plus anciens, autrement l'outil de conception de remplacement, Vivado Design Suite.

ISE permet au développeur de synthétiser ("compiler") leurs conceptions, d'effectuer une analyse temporelle, d'examiner des diagrammes RTL, de simuler la réaction d'une conception à différents stimuli et de configurer l'appareil cible avec le programmeur. Les autres composants livrés avec Xilinx ISE incluent le kit de développement intégré (EDK), un kit de développement logiciel (SDK) et ChipScope Pro. L'ISE Xilinx est principalement utilisé pour la synthèse et la conception de circuits, tandis que l'ISIM ou le simulateur logique ModelSim est utilisé pour les tests au niveau du système.

Comme cela est couramment pratiqué dans le secteur de l'automatisation de la conception électronique commerciale, Xilinx ISE est étroitement couplé à l'architecture des propres puces de Xilinx (dont les composants internes sont hautement propriétaires) et ne peut pas être utilisé avec les produits FPGA d'autres fournisseurs. Compte tenu de la nature hautement propriétaire des gammes de produits matériels Xilinx, il est rarement possible d'utiliser des alternatives open source aux outils fournis directement par Xilinx, bien qu'à partir de 2020, certaines tentatives exploratoires soient en cours.

Le logiciel xilinx ISE est un logiciel de description, de simulation, et de programmation de circuits et systèmes numériques sur des composant programmables. Le logiciel ISE possède

une version gratuite et téléchargeable du site de Xilinx (www.xilinx.com). La suite ISE permet :

- La description de circuits numériques sous forme de schémas logiques, de machines à états finis ou en langages de description matériel (VHDL, Verilog, ABEL),
- La compilation, la simulation comportementale,
- La synthèse, le placement routage et l'implémentation,
- La simulation temporelle et l'analyse de timing,
- La programmation sur les circuits programmables de Xilinx (CPLD et FPGA).

➤ **Simulation :**

Les tests au niveau du système peuvent être effectués avec ISIM ou le simulateur logique ModelSim, et ces programmes de test doivent également être écrits en langages HDL. Les programmes de banc d'essai peuvent inclure des formes d'onde de signal d'entrée simulées ou des moniteurs qui observent et vérifient les sorties du dispositif testé.

ModelSim ou ISIM peuvent être utilisés pour effectuer les types de simulations suivants :

- Vérification logique, pour s'assurer que le module produit les résultats attendus
- Vérification comportementale, pour vérifier les problèmes de logique et de synchronisation
- Simulation Post-place & route, pour vérifier le comportement après le placement du module dans la logique reconfigurable du FPGA

4.4 Description de programme vhdl gener à partir de matlab :

4.4.1 Présenter le programme Matlab du massive-MIMO precoding utilisé et l'expliquer le système Massive MIMO simulé

Le système qui a été utilisé, c'est un system massif MIMO avec beamforming du type Maximum Likelihood (une méthode d'estimation des paramètres d'une distribution de probabilité supposée, compte tenu de certaines données observées), avec une station de base qui emploie 64 antennes avec 4 utilisateurs, chacun ayant 16 antennes.

```

clear;clc;
% ----- System Parameters -----
Num_users=4; % Number of users
TX_ant=64; %Number of UPA TX antennas
TX_ant_w=sqrt(TX_ant); % width
TX_ant_h=sqrt(TX_ant); % hieght ind_TX_w=reshape( repmat([0:1:TX_ant_w-1],TX_ant_h,1),1, TX_ant_w);
ind_TX_h=repmat([0:1:TX_ant_h-1],1, TX_ant_w);

RX_ant=16; %Number of UPA RX antennas
RX_ant_w=sqrt(RX_ant); % width
RX_ant_h=sqrt(RX_ant); % hieght
ind_RX_w=reshape( repmat([0:1:RX_ant_w-1],RX_ant_h,1),1, RX_ant_w*RX_ant_h);
ind_RX_h=repmat([0:1:RX_ant_h-1],1, RX_ant_w);

% ----- Channel Parameters -----
Num_paths=10; %Number of channel paths

% ----- Simulation Parameters -----
SNR_dB_range=-20:3:10; % SNR in dB
Rate_DG_ZF=zeros(1,length(SNR_dB_range)); % Carrying the rate of the fully digital ZF precoder

ITER=500; % Number of iterations

% ----- Simulation starts -----
for iter=1:1:ITER
    % Generate user channels
    [H,a_TX,a_RX]=generate_channels(Num_users,TX_ant_w,TX_ant_h,RX_ant_w,RX_ant_h,Num_paths);
    % H is a 3-dimensional matrix, with Num_users,RX_ant,TX_ant dimensions

    % Stage 1 of the proposed algorithm (Analog precoding)

```

4.4.2 Procédure pour générer du code VHDL avec matlab

Le VHDL Génère automatiquement du code HDL à partir de MATLAB et permet aux ingénieurs d'implémenter des conceptions sur des FPGA et des ASIC à l'aide du langage MATLAB largement utilisé. MathWorks a également introduit HDL Vérifier, qui inclut des capacités hardware-in-the-loop (HIL) pour les FPGA qui peuvent être utilisées pour tester des conceptions sur des FPGA et des ASIC. Avec ces deux produits, MathWorks fournit désormais la génération et la vérification de code HDL à l'aide de MATLAB By Simulink. HDL Coder génère du code VHDL et Verilog portable et synthétisable à partir de fonctions MATLAB et de modèles Simulink qui peuvent être utilisés pour programmer des FPGA ou pour prototyper et concevoir des ASIC. En conséquence, les équipes d'ingénierie peuvent désormais identifier

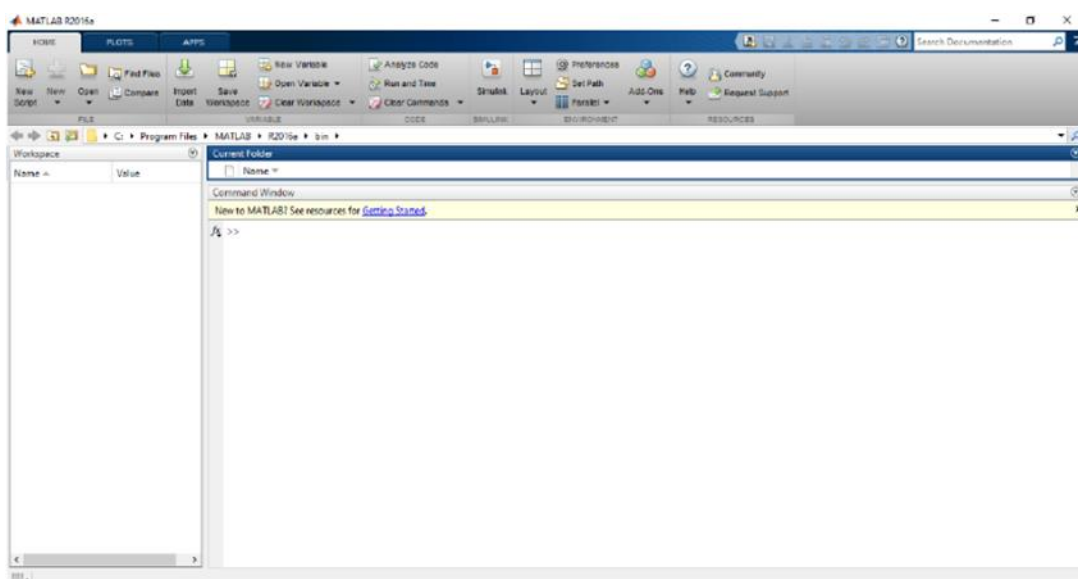
Immédiatement le meilleur algorithme pour la mise en œuvre matérielle. HDL Vérifier prend désormais en charge la vérification HIL pour les FPGA pour les cartes FPGA Altera et Xilinx. HDL

Vérifier fournit des interfaces de Co simulation qui relie MATLAB et Simulink avec des simulateurs

HDL tels que Cadence Incisive, Mentor Graphiques ModelSim et Questa. Grâce à ces fonctionnalités, les ingénieurs peuvent vérifier rapidement si leur implémentation HDL correspond aux algorithmes MATLAB et aux spécifications du système Simulink.

La procédure pour générer du code **VHDL** avec Matlab est la suivante :

1. ouvrir Matlab

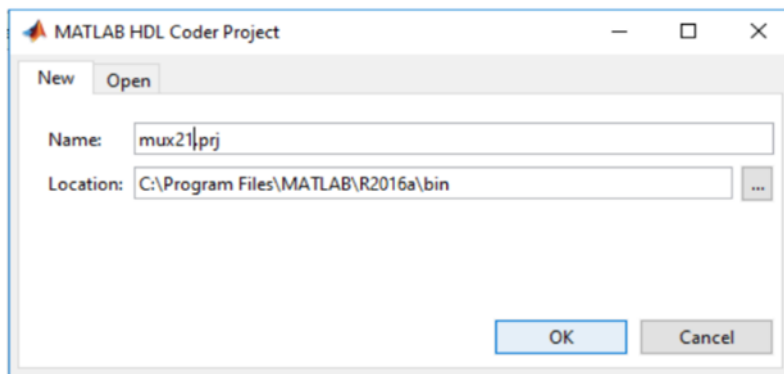


2.Écrivez une fonction que vous souhaitez convertir en VHDL

3. Allez sur Apps et recherchez HDL coder



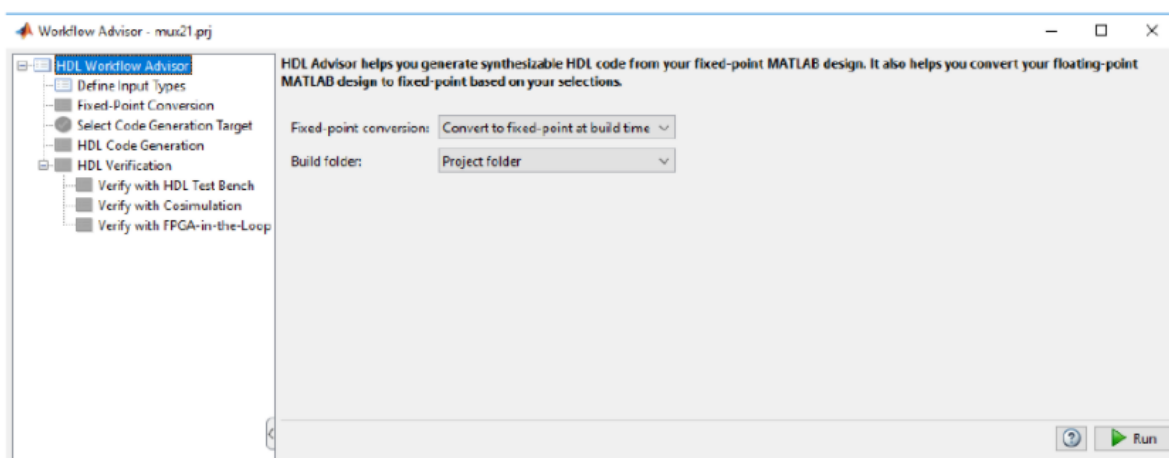
4.Ouvrez HDL Coder et faire new project



5. Add MATLAB function file

6. Go to Workflow Advisor

7. Sélectionnez Convert to fixed-point at build time car la double opération ne convient pas à FPGA



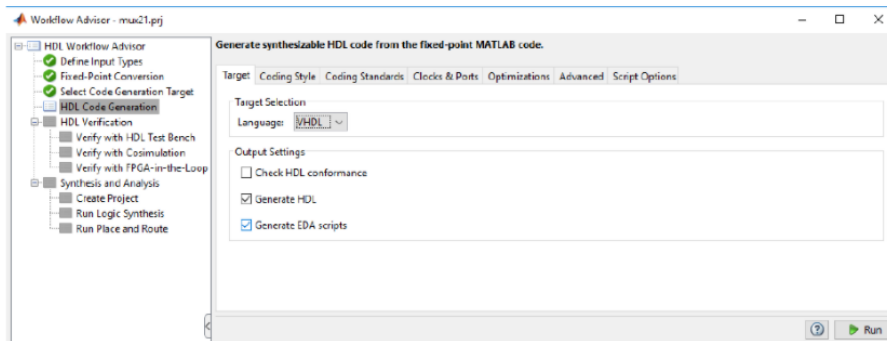
9. allez a Fixed-point conversion and define Static Min and Static Max

10. Compute Derived Ranges

11. Validate Types

12. Select Code Generation Target

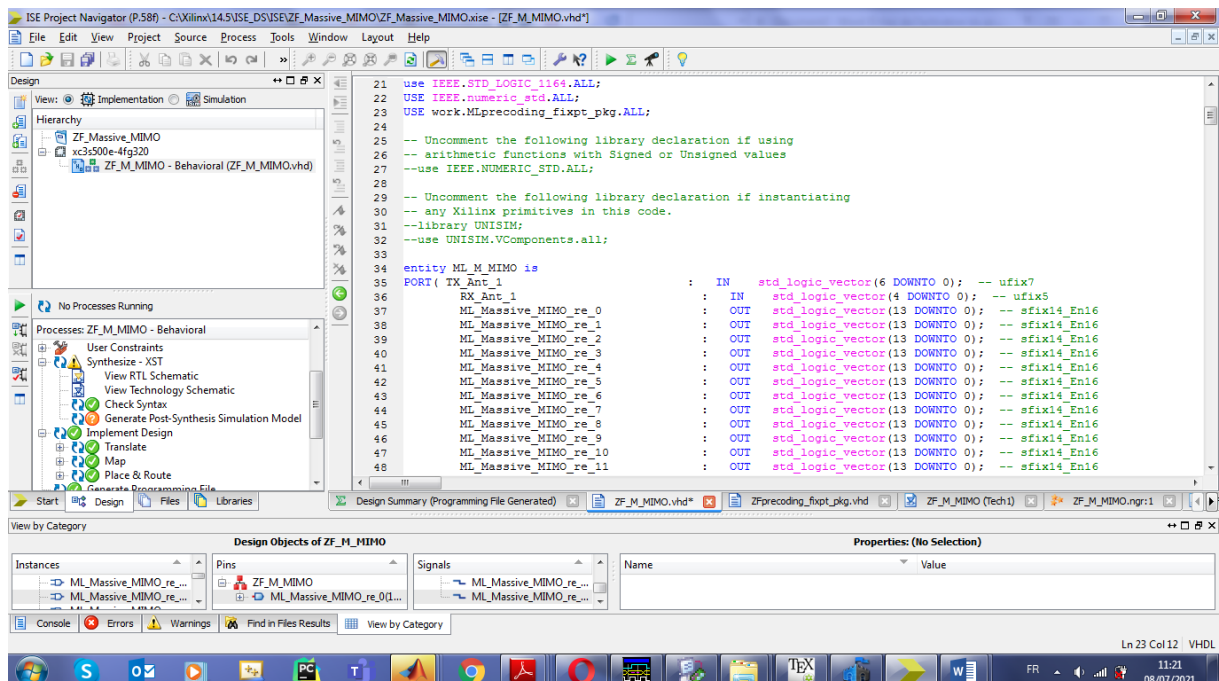
13. HDL Code Generation – Select VHDL



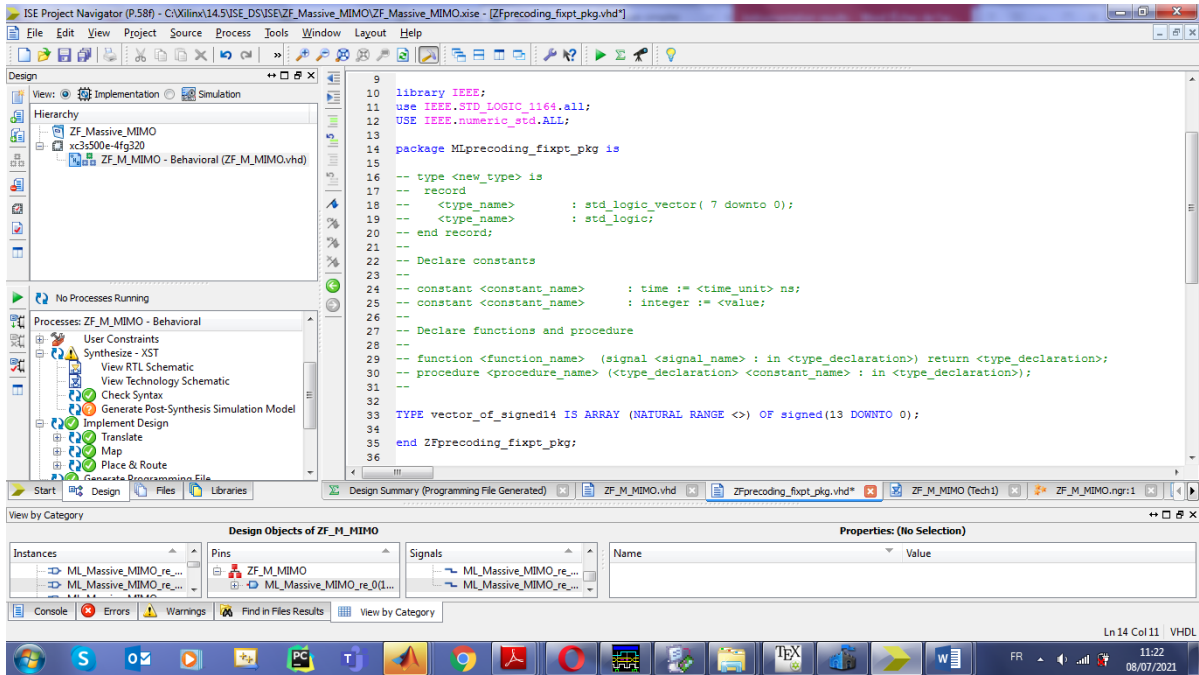
14. Run.

4.4.3 Présenter le programme VHDL obtenu

Le programme VHDL contient une entité nommée ML_M_MIMO avec deux ports d'entrer TX_antenne et RX_antenne et les autres sont des sorties. On donne à l'entité l'information sur l'antenne d'émission et de réception après cette procédure l'entité fait le calcul et génère le beamforming du type ML



C'est un package qui est écrit dans le premier programme puisqu'il s'intéresse au package pour générer le pre-cogade numérique.



4.4.4 Présenter le schéma RTL obtenu (synthétiseur)

La figure (4.3) c'est l'entité qui a été présenté sur le programme VHDL par ML_M_MIMO

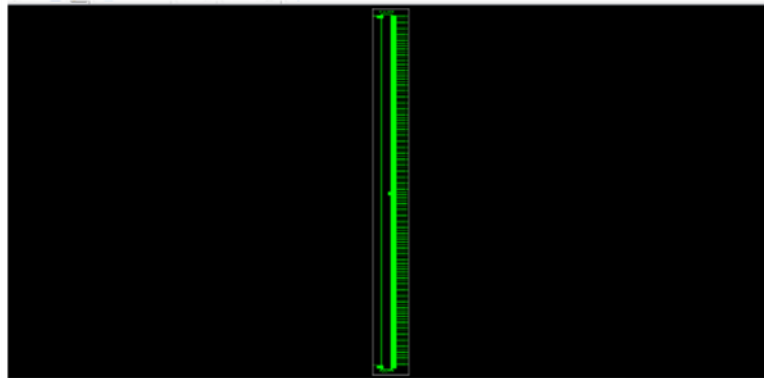


Figure 4.3:Entité

CONCLUSION GÉNÉRALE

La technique MIMO massive est considérée par l'industrie comme une technique clé dans la future 5 G. Il peut améliorer considérablement la capacité du canal, l'efficacité énergétique et l'efficacité du spectre du système de communication sans fil.

Le but du travail présenté dans ce mémoire, est l'implémentation d'un système MIMO massive sur une carte FPGA du type Spartan-S3. Nous avons structuré notre mémoire en quatre chapitres. Dans le premier chapitre, nous avons mené une description sur les différentes parties d'un système de communication numérique.

D'abord, nous avons présenté le fonctionnement d'une chaîne de transmission numérique, de la source d'information binaire au destinataire, par les étapes successives de codage, de modulation, de transmission dans un canal physique et de démodulation. Ensuite, nous présenterons les différents types des canaux de transmission puis le récepteur avec ses paramètres.

Le deuxième chapitre nous avons mené une étude sur la technique MIMO, nous avons aussi abordé dans ce chapitre la technique MIMO massive. Passer revue les différents types du système, le modèle de canal, la capacité du système, les codes spatio-temporels (STBC, STTC). Ensuite, nous avons vu la technique MIMO massive avec le modèle de canal mmwave et les techniques de pré codages. Enfin, nous avons fait une analyse d'une simulation des systèmes de pré-codage Massive MIMO avec la technique Digital pre-coding ZF en fonction de SNR. ET à partir de là, nous avons conclu que lorsqu'on ajoute des antennes de puissance, l'efficacité spectrale augmente.

Pour le troisième chapitre, nous avons exposé les réseaux logiques programmables PLD avec ses différents types, par la suite, nous avons vu la technique FPGA avec la structure générale d'un circuit FPGA et FPGA de la famille Xilinx. Enfin, nous avons fait un résumé sur le langage VHDL.

La quatrième partie du travail était consacrée à l'analyse des résultats de simulation à l'aide de logiciels ISE Xilinx, nous avons commencé par la description de la carte FPGA Xilinx spartan S3 et logiciel ISE Xilinx. Ensuite, nous avons fait une description et une explication pour la procédure du programme vhdl générer à partir de Matlab. Enfin, une analyse de la simulation du programme VHDL sous ISE et synthétiseur généré.

Bibliographie

- [1] : O. Berder. « Optimisation et Stratégies d'Allocation de Puissance des Systèmes de Transmission MultiAntennes » thèse pour obtenir le diplôme de doctorat en Électronique, l'université –Bretagne Occidentale ,2002.
- [2] : Ben Abdallah. « Effet d'entrelacement sur turbo égalisation MAP » Mémoire de fin d'étude pour l'obtention du diplôme de Master 2 l'Université-Dr Moulay Taher-Saida ,2012.
- [3] : R. Ghalem et K. Hafsi. « Analyse du système de transmission radio mobile MC-DS-CDMA » Mémoire de fin d'étude pour l'obtention du diplôme de Master 2 en Télécommunication, l'Université-Dr Moulay Taher-Saida ,02 Juin 2016.
- [4] : A. Glavieux et M. Joindot. « Introduction aux Communications numériques » Edition Masson France 1996.
- [5] : I. Mohammedi et F. Amara. « Analyse des propriétés des décodeurs M-BCJR » Mémoire de fin d'étude en vue de l'obtention du diplôme d'Ingénieur d' Etats en Télécommunication, l'Université-Dr Moulay Taher-Saida, Juin 24/06/2009
- [6] : L. Hacini. « Schémas hybrides d'acquisition adaptative de codes PN pour des communications DSCDMA dans un canal Rayleigh » thèse pour obtenir le diplôme de doctorat en Électronique, l'université Mentouri-Constantine, Juin 2012.
- [7] : S. Berhab. « Algorithmes adaptatifs pour des transmissions multiantennes MIMO haut débit en réseaux sans fil » Mémoire pour l'obtention du diplôme de Magister en Télécommunications et Informatique Spatiales, l'Université-Abou Bakr Belkaid-Tlemcen, 2011.
- [8] : M. Fassi Benattou. « Contribution à l'étude des codes ZCZ (Zero Correlation Zone) : Application au système CDMA » thèse pour obtenir le diplôme de doctorat en Électronique l'Université- Djillali Liabès - Sidi- Bel-Abbes
- [9] : <https://slideplayer.fr/slide/11884622/>
- [10] : <https://www.becoz.org/these/memoirehtml/ch05s02.html>
- [11] : C. Khalfallah. « L'égalisation dans les systèmes multiple Input – Multiple output « MIMO » » Mémoire de fin d'étude pour l'obtention du diplôme de Magistere en Electronique, l'Université- Mohamed Boudiaf-Oran ,2010.
- [12] : DEHEMCHI MARWA BOUHAFER ASMA (thèse de master) Etude et Analyse de Méthodes de Beamforming pour des Systèmes Multi-utilisateurs Pour la 5G
- [13] : Nimay Ch Giri, Anwasha Sahoo, J. R. Swain, P. Kumar, A. Nayak, P. Debogswami, Lecturer, Department of ECE,B.Tech Scholar, Centurion University of Technology and Management, Odisha, India, Capacity & Performance Comparison of SISO and MIMO System for Next Generation Network (NGN), International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 9, September 2014
- [14] : Kritika Sengar, Nishu Rani, Ankita Singhal, Dolly Sharma, Seema Verma, Tanya Singh,

Banasthali University, Newai, India, Amity Institute of Information and Technology University, Noida, India, Study and Capacity Evaluation of SISO, MISO and MIMO RF Wireless Communication Systems, International Journal of Engineering Trends and Technology (IJETT) – Volume 9 Number 9 - Mar 2014.

[15] : Antoine Roze (thèse doctorat) Massive MIMO, une approche angulaire pour les futurs Systèmes multi-utilisateurs aux longueurs d'onde millimétriques.

[16] : Stephen Boyd, Symmetric matrices, quadratic forms, matrixnorm, and SVD, Stanford university, EE263 Autumn 2007-08. Review of Linear Algebra.

[17] : * Y. Liu, Y. Yu, W.-J. Lu, H.B. Zhu, Stochastic multiple-input multiple-output channelmodel based on singular value decomposition, IET Commun. 9 (15) (2015) 1852–1856.

* G. Lebrun, J. Gao, M. Faulkner, MIMO transmission over a time-varying channel usingSVD, IEEE Trans. Wirel. Commun. 4 (2) (2005) 757–764.

[18] : S. P. Jadhav, andV. S. Hendre, “Performance of Maximum Ratio Combining (MRC) MIMO Systems for Rayleigh Fading channels,” International Journal of Scientific and Research Publications, vol 3, pp. 2250-3153, February 2013.

[19] : Irfan Ahmed, Hedi Khammari, Adnan Shahid, Ahmed Musa, Kwang Soon Kim, Eli De Poorter, Ingrid Moerman, “A Survey on Hybrid Beamforming Techniques in 5G: Architecture and System Model Perspectives,” Journal of Latex Class Files, Vol.XX, NO. X, avril 2021

[20] : Shahid Mumtaz, Jonathan Rodriguez and Linglong Dai, “mmWave Massive MIMO: A Paradigm for 5G,” © Elsevier Inc. Avril 2021

[21] : IEEE Globecom 2015 conference by gift from Huawei Technologies Co. Ltd.avril 2021

[22] : Tewelgn Kebede Engda , Yihewew Wondie and Johannes Steinbrunn, «Massive MIMO, mmWave and mmWave-Massive MIMO Communications: Performance Assessment with Beamforming Techniques » .Avril 2021

[23] : A. KILANI, Introduction aux circuits Logiques Programmables.chap 05.mai 2021.

[24] : S.T.S. GRANVILLE Philippe LETENNEUR, LES CIRCUITS LOGIQUES PROGRAMMABLES.mai 2021

[25] : Dr. MOHAMMED ZAKARYA BABA-AHMED, FPGA et programmation VHDL [Université Hassiba Benbouali de Chlef master1]

[26] : Samitiana RAMBOAMAMPIANINA, Exploitation de la technologie FPGA pour générer les signaux de commande des onduleurs autonomes triphasés[UNIVERSITE D'ANTANANARIVO ECOLE SUPERIEURE POLYTECHNIQUE master 2].

[27] : Jerry TEKOBON. FPGA and Traffic Network Analysis, Ecole Nationale Supérieure Polytechnique de Yaounde Cameroun - Ingénieur de Conception en Génie Electrique et Télécommunication 2007.,Chap 2.

[28] : Samitiana RAMBOAMAMPIANINA, Exploitation de la technologie FPGA pour générer les signaux de commande des onduleurs autonomes triphasés[UNIVERSITE D'ANTANANARIVO ECOLE SUPERIEURE POLYTECHNIQUE master 2].

[29] : Jerry TEKOBON. FPGA and Traffic Network Analysis, Ecole Nationale Supérieure Polytechnique de Yaounde Cameroun - Ingénieur de Conception en Génie Electrique et Télécommunication 2007.,Chap 2.

[30] : www.ece.ucdavis.edu/~soheil/teaching/EEC180B-S06/ds099-1.pdf?q=ds099
consulté le 12/08/2021 a 15 :15

[31] : architectural<https://www.ece.ucdavis.edu/~soheil/teaching/EEC180B-S06/ds099-1.pdf?q=ds099> consulté le 10/07/2021 a 10 :15