# Chapitre IV

# Architecture numérique et générique de modulation et codage pour la transmission de données

## 1. Introduction

***Dans le chapitre I nous avons montré que les différences essentielles entre les différents standards et normes de télécommunication sont les types de modulations et les codages utilisés. En effet, la modulation employée, peut varier entre une modulation de phase (M-PSK), de fréquence (FSK, GFSK), ou une modulation d'amplitude et de phase (M-QAM). De la même manière, les codes détecteurs et correcteurs d'erreurs peuvent aller du simple code CRC aux codes les plus complexes (code convolutif, Reed-Solomon, Turbo Codes).***

***Le développement des architectures génériques peut générer plusieurs types de modulation dont l'intérêt touchera les systèmes RL et RC, ce qui est un objectif très intéressant. Dans ce chapitre, après une synthèse des architectures numériques pour la modulation et le codage canal, nous proposons d'abord une architecture générique et universelle pour la modulation numérique en utilisant la technique de paramétrisation.***

***Ensuite, nous proposons une nouvelle architecture universelle utilisée pour quatre fonctionnalités; la démodulation numérique des signaux modulés en phase et en fréquence, la génération de code de brouillage connue par le code Gold et le codage de canal.***

## 2. Traitement numérique vs traitement analogique

Dans les chapitres I et II, nous avons remarqué que les couches les plus basses de la chaîne de transmission sont les plus sensibles aux changements de protocoles de transmission ou du standard, car elles doivent produire un signal dont les caractéristiques physiques et temporelles sont très dépendantes des spécifications du canal de communication ou du standard ciblé.

Habituellement, la plupart des architectures de la couche physique repose sur des composants électroniques analogiques. Bien que les composants analogiques soient compatibles avec les hautes fréquences de fonctionnement, leurs champs d'application et les possibilités d'évolution restent très limités. Par conséquent, les composants analogiques sont peu adaptés à l'évolution des normes, des algorithmes et des protocoles utilisés.

Plusieurs tentatives ont été effectuées pour transposer sous forme de circuits numériques les solutions technologiques ayant fait leurs preuves dans le domaine analogique [107] [108]. Cependant, la transposition des fonctionnalités analogiques en numérique introduit des chemins critiques difficiles à réduire, qui limitent la fréquence de fonctionnement et restreignent l'emploi de tels systèmes aux applications à faible débit. C'est notamment le cas pour les boucles à verrouillage de phase (PLL), présentes dans la plupart des modulateurs et démodulateurs analogiques, et dont les fonctionnalités sont très difficiles à transposer efficacement en numérique [93] [109].

Une nouvelle méthodologie proposée [109] consiste à faire une traduction directe de la fonctionnalité souhaitée en un algorithme adapté aux circuits numériques, sans passer par une étape analogique susceptible d'introduire des contraintes en contradiction avec la démarche de conception de ces circuits. Dans ce qui suit, nous proposons des architectures génériques pour la modulation et démodulation numérique en suivant la technique de la paramétrisation et la méthodologie proposée dans [109].

## 3. Architectures génériques pour la modulation numérique

Dans ce paragraphe nous utilisons une architecture M-QAM donc la densité de la modulation du signal transmis peut varier facilement, puis sur la base de cette architecture, nous proposons une autre architecture générique numérique pour la modulation de phase, la modulation de fréquence et la modulation QAM.

### 3.1.Un modulateur numérique M-QAM

La modulation QAM présente la particularité de nécessiter le contrôle de deux paramètres de la sinusoïde de sortie, à savoir, sa phase et son amplitude [9]. Un signal s(t) de QAM, est représenté par l'équation (1):

$$s(t) = A_k \cos(\varphi_k) \cos(\omega t) - A_k \sin(\varphi_k) \sin(\omega t) = I\, cos(\omega t) + Q sin(\omega t) \quad (1)$$

où $I$ et $Q$ sont les composants in-phase et quadrature respectivement, $\omega$ est la fréquence de la porteuse, $A_k$ $et$ $\varphi_k$ sont respectivement l'amplitude et la phase d'un signal QAM.

Un modulateur numérique QAM est composé de cinq parties principales (figure IV.1) [110][111]:

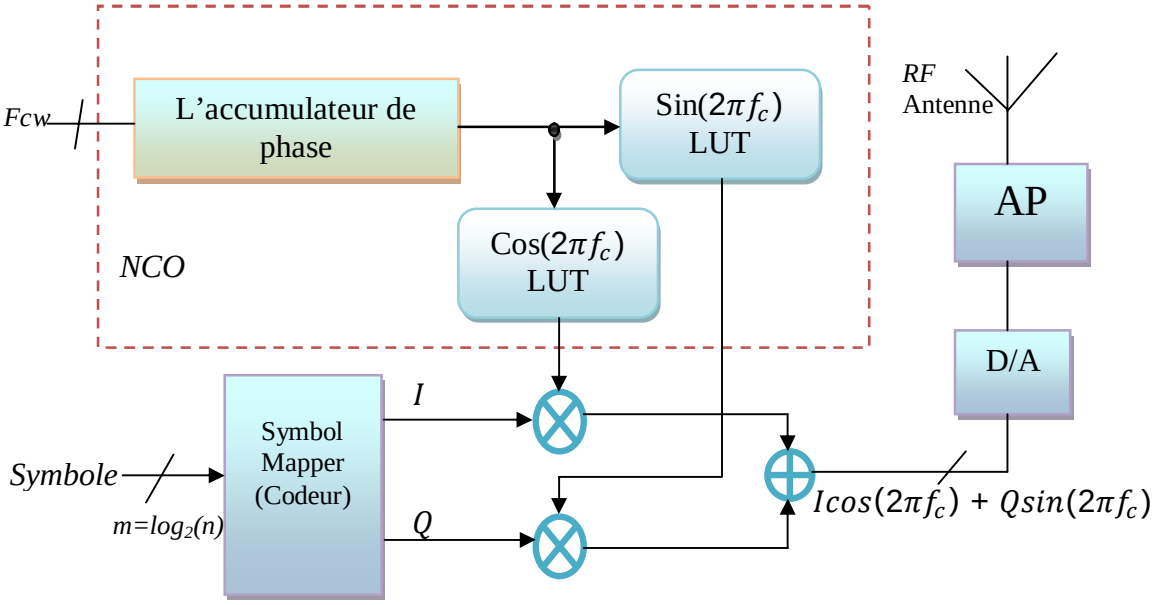


Figure IV. 1. Structure d'un modulateur QAM numérique.

**– le codeur (Symbol Mapper)**: en groupant, sous forme d'un bloc $n$ symboles binaires indépendants, on obtient un alphabet de $M = 2^n$ symboles M-aires. Ainsi un symbole M-aire véhicule l'équivalent de $n = \log_2 M$ bits. Par exemple, pour la modulation QAM-16, chaque symbole représente 4 bits. Généralement, on utilise le codage de Gray, de tel sorte que chaque deux symboles successifs se diffèrent par un seul bit. Ensuite, le codeur (Symbol Mapper) génère les deux composants in-phase (I) et quadrature (Q) correspondant de ce symbole.

**– l'accumulateur de phase** dont le rôle est de produire le terme de phase $\omega t$. Ce terme correspond à la phase instantanée de l'échantillon de la sinusoïde produit par le modulateur (parfois on ajoute à ce terme une phase constant $\varphi_0$). En effet, le principe de l'accumulateur de phase consiste à *recréer numériquement*, point par point, le signal que l'on désire obtenir. L'architecture de l'accumulateur de phase est représentée dans la figure IV.2.

L'accumulateur de phase est piloté par un mot de contrôle de la fréquence $F_{cw}$ correspondant à l'incrément de phase appliqué à chaque période d'horloge du système.

L'incrément de phase est ajouté à l'incrément de phase résultant pour chaque période d'horloge du système.

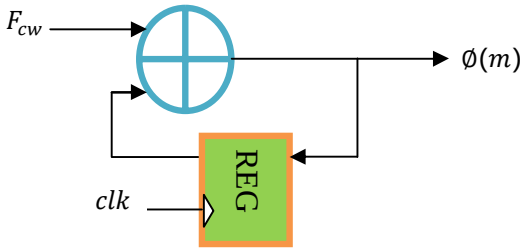


Figure. IV.2. Architecture de l'accumulateur de phase

« $F_{cw}$ » détermine la fréquence $f$ effective résultante du signal modulé, en fonction de la fréquence d'horloge $f_{clk}$ de l'architecture hôte. Le $m^{e}$ mot de phase produit par cet étage vaut :

$$\phi(m) = m \cdot Fcw. \qquad (2)$$

**– le générateur numérique de fréquence** qui permet de produire les échantillons numériques de sinusoïdes (cosinus et sinus) à partir des paramètres que lui fournit l'accumulateur de phase. Le générateur numérique de fréquence joue le rôle d'un oscillateur à commande numérique (NCO). Généralement, il est implémenté par deux tables de correspondance LUT (ROM), dans lesquelles sont stockés les échantillons pré-calculés de la fonction cosinus et sinus. En effet, dans la mémoire sont stockées N valeurs correspondantes à une période du signal. La lecture se fait par incrément $F_{cw}$ pour permettre de sélectionner l'adresse de la valeur désirée. Puis par une lecture successive de ces valeurs, on peut restituer le signal désiré en sortie [93]. On obtient un signal sinusoïdal de fréquence [111] :

$$f_s = \frac{f_{clk} \cdot F_{cw}}{N} \qquad (3)$$

**– deux multiplieurs** sont utilisés pour faire la multiplication de la composante in-phase par le cosinus issue de la première LUT, et la composante quadrature par le sinus issue de la deuxième LUT (les deux composantes I et Q sont filtrées par des filtres FIR avant d'être multiplié). Les produits sont finalement ajoutés afin de générer un signal modulé QAM conformément à l'équation (1).

**– le convertisseur numérique analogique** est le dernier élément de la chaîne de traitement numérique, dont le rôle est de convertir les échantillons en signaux adaptés au médium de transmission.

## 3.2. Architecture proposée pour un modulateur QAM numérique

Dans l'architecture numérique pour la modulation QAM de la figure IV.1, on remarque deux inconvénients:

- l'architecture (figure IV.1) utilise deux multiplieurs complexes pour la multiplication de la composante in-phase par le cosinus, et de la composante quadrature par le sinus. Malheureusement, la consommation d'un multiplieur complexe en ressource est grande dans le FPGA.
- l'architecture utilise des LUT (ROM), pour la génération de la porteuse (le signal sinus et cosinus). Pour obtenir une bonne précision, à la fois en phase et en amplitude de la porteuse, il est nécessaire d'utiliser une ROM de grande taille dont les conséquences sont néfastes sur la fréquence d'horloge maximale du circuit et sur la surface occupée [93].

Dont le but de réaliser une architecture pour la modulation QAM à fréquence de fonctionnement très rapide, moins occupante en surface sur FPGA et à usage universel, nous avons proposé une nouvelle architecture QAM numérique et générique représentée par la figure IV.3. La nouvelle architecture est basée totalement sur l'équation (4) qui représente un signal modulé QAM:

$$s(t) = A_k.cos(\omega t + \varphi_k) = A_k.cos(m \cdot F_{cw} + \varphi_k) \quad (4)$$

Où $A_k$ $et$ $\varphi_k$ sont respectivement l'amplitude et la phase correspondant au symbole $'k'$.
En effet l'équation (1) représente le développement de l'équation 4 car :

$$s(t) = A_k\, cos(\omega t + \varphi_k) = A_k \cos(\varphi_k) \cos(\omega t) - A_k \sin(\varphi_k) \sin(\omega t) = I\, cos(\omega t) + Q sin(\omega t) \quad (5)$$

avec aussi $A_k = \sqrt{(I^2 + Q^2)}$ et $\varphi_k = arctan\left(\frac{Q}{I}\right)$.

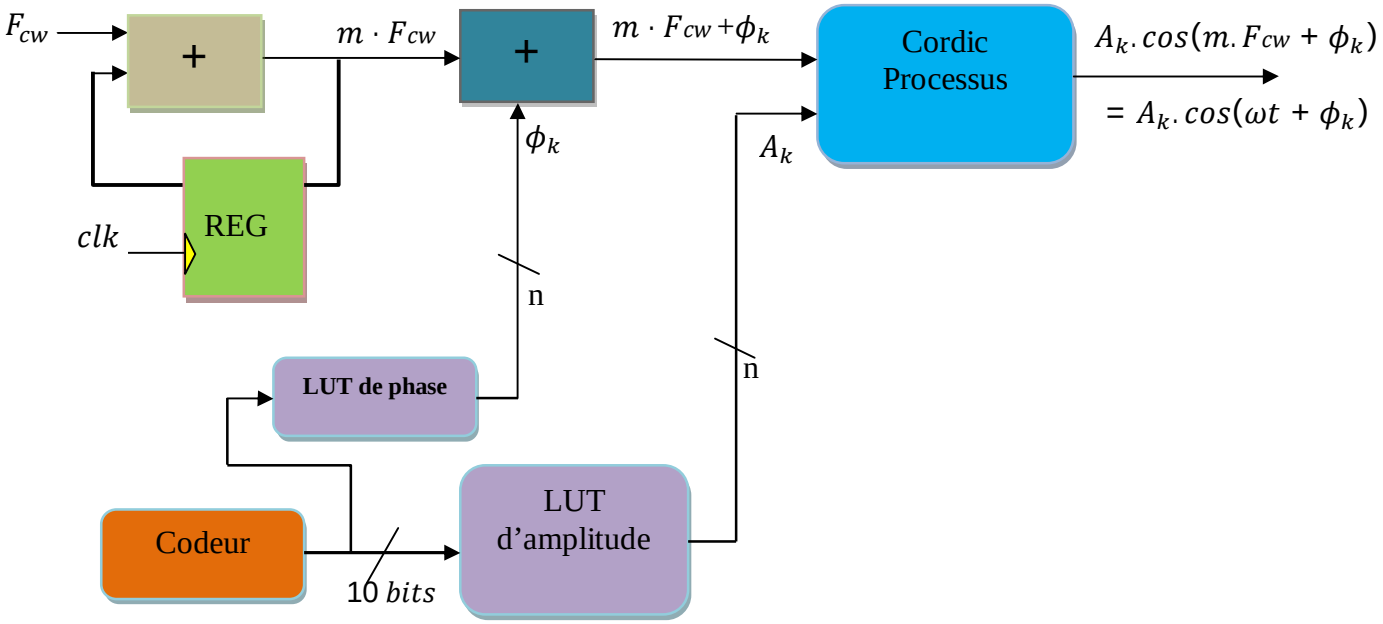


Figure IV.3. Architecture du modulateur numérique QAM proposé.

Pour comprendre le fonctionnement de cette architecture, nous détaillons ci-dessous les principales modifications ainsi que le principe de fonctionnement des composantes constituant l'architecture IV.3.

### 3.2.1. Le Codeur

Pour mettre une architecture de modulation M-QAM universelle qui doit s'adapter facilement à plusieurs types de constellation, nous avons implémenté un codeur générique. Nous avons alors choisi la modulation de type QAM 1024 comme une base de représentation des autre types de modulation (QAM 512, QAM 256….QPSK). La première étape du codage consiste en la conversion série / parallèle sur 10 bits du signal en entrée, car le QAM 1024 est de n = 10 bits ($M = 2^n = 2^{10} = 1024$). Puis en fonction du type de modulation choisi, les données sont converties de manière à pouvoir être placées sur une constellation de type QAM 1024. Pour pouvoir placer les autres types de modulation dont leur nombre de bits est inférieur à 10 (par exemple le QAM 16 est de 4 bits seulement), on insère des bits nuls (zéro) pour compléter la trame de 10 bits. Ces modifications sont représentées sur la figure IV.4.

De plus, le placement des bits dans la trame utilise les propriétés de symétrie des différentes constellations [4] [90]. En effet, si on regarde l'exemple de la figure IV.5 pour la modulation 16-QAM, on remarque que la répartition des points dans les différents quadrants de la constellation peut être obtenue simplement par une rotation d'un angle multiple de π/2 du quadrant principal.

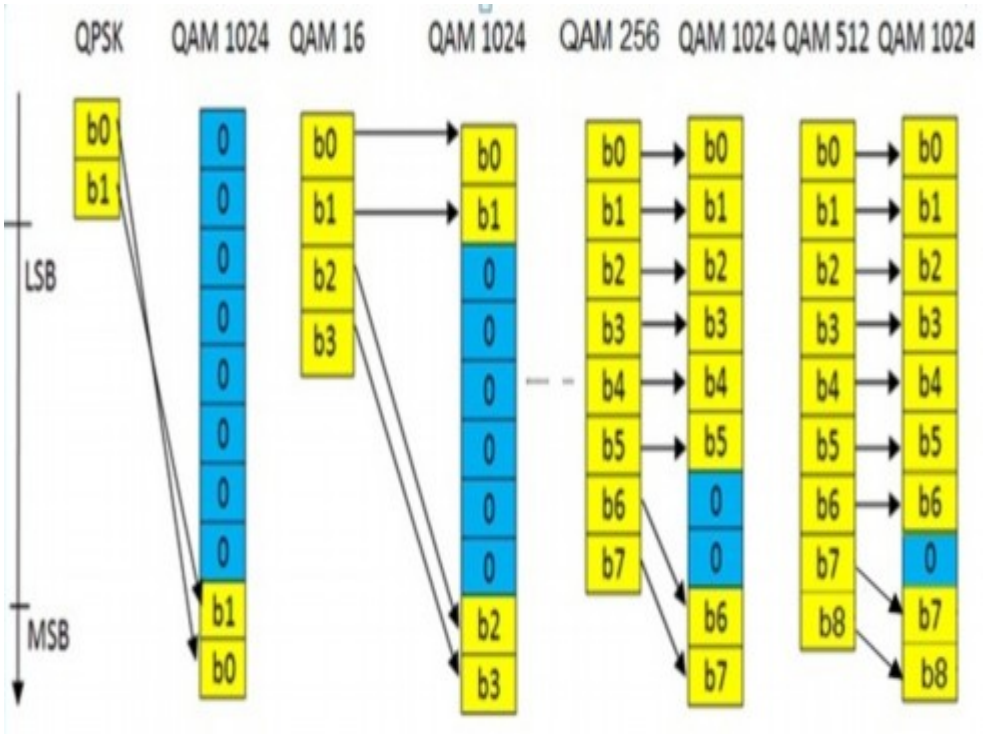


Figure IV.4. Conversion des trames au format QAM 1024.

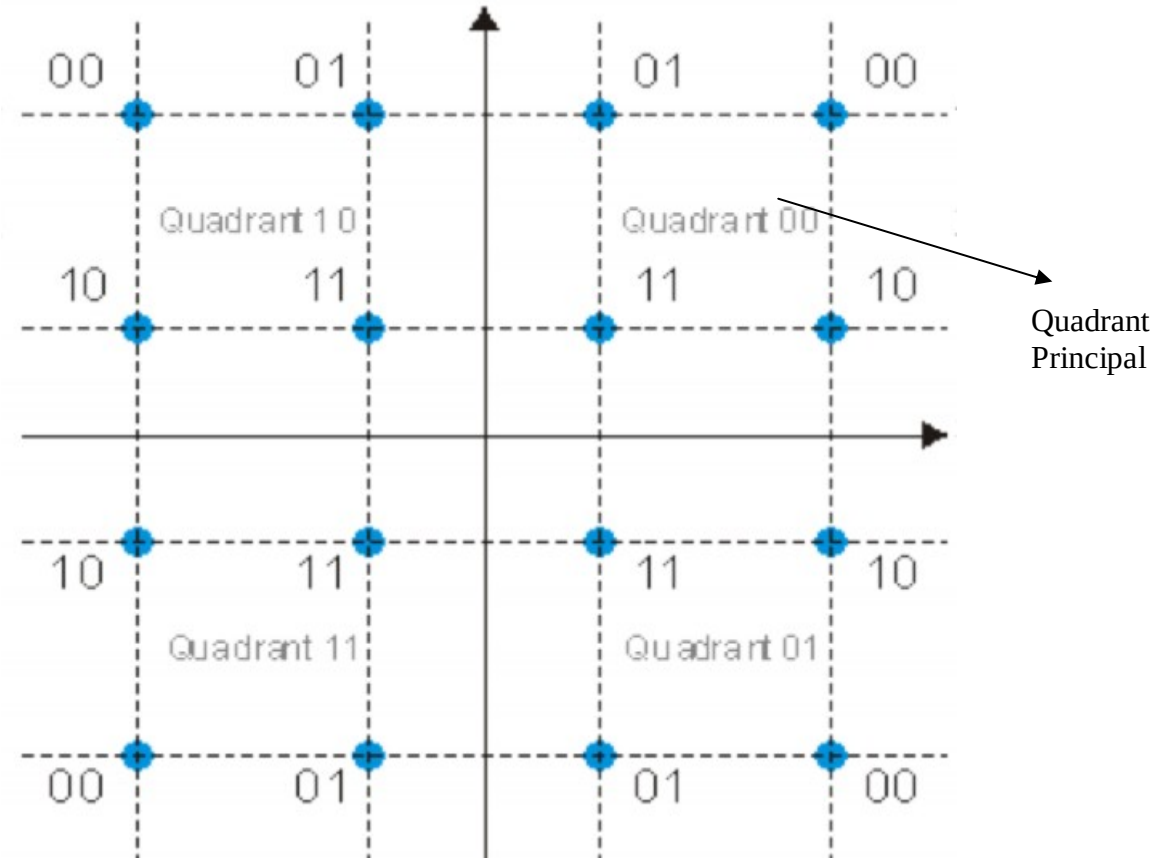


Figure IV.5. Propriété de symétrie dans la constellation QAM.

Ainsi, on remarque que les deux bits de poids fort de chaque symbole représentent le quadrant dans lequel il va se positionner:

– quadrant 1 (α = 0): bits de poids fort 00;

– quadrant 2 (α = +π/2): bits de poids fort 01;

– quadrant 3 (α = +π): bits de poids fort 11;

– quadrant 4 (α = − π/2): bits de poids fort 10.

En sortie du codeur, nous obtenons donc un bus véhiculant des trames QAM1024 à 10 bits, indépendamment du type de modulation souhaité en entrée.

### 3.2.2. Le LUT d'amplitude et le LUT de phase

La sortie de ce codeur est utilisée comme adressage pour deux LUTs. La première LUT génère la phase $\varphi_k$ qui correspond à chaque symbole $k$, et la deuxième LUT génère en fonction de cette valeur d'adressage l'amplitude $A_k$. Dans notre architecture proposée, nous avons donc généré directement la phase $\varphi_k$ et l'amplitude $A_k$ au lieu de générer les deux composants in-phase 'I' et quadrature 'Q'. Ceci nous permet donc de s'affranchir de

l'utilisation des deux multiplieurs complexes comme dans le cas de la figure (IV.1). La phase $\varphi_k$ et l'amplitude $A_k$ obtenues, nous permettent de générer directement le signal QAM en utilisant l'équation 4.

Puisque l'information sur le quadrant est véhiculée par les deux bits de poids fort de la trame QAM 1024, seuls les huit autres bits seront utiles pour déterminer la valeur de l'angle correspondant à la phase. Ainsi, nous avons développé un petit programme pour générer directement la valeur de l'angle correspondant à la phase.

Pour les valeurs des amplitudes, nous avons aussi développé un petit programme qui génère automatiquement une matrice contentant les valeurs nécessaires.

### 3.2.3. L'accumulateur de phase

Le rôle de l'accumulateur de phase est de produire le terme de phase ω*t*. Si on regarde l'équation (2) qui donne le $m_e$ mot de phase produit par l'étage de l'accumulateur de phase, on remarque que celle-ci produit les valeurs suivantes: $F_{c\omega 0}$, $F_{c\omega 1}, \ldots . F_{c\omega n}$. où $F_{cw1} = 2F_{cw0}$, $F_{cw2} = 3F_{cw0} \ldots F_{cwn} = mF_{cw0}$.

Le $m_e$ mot de phase produit par l'étage de l'accumulateur de phase correspond au terme « $\omega t$». Enfin pour produire le terme $(\omega t + \varphi_k)$, on ajoute la phase « $\varphi_k$ » qui correspond au symbole transmit $k$ . Pour cette dernière opération, on utilise un additionneur rapide. Il y a plusieurs architectures des additionneurs, mais il faut noter que plus une architecture permet d'effectuer le calcul rapidement, plus sa surface augmente. C'est pour cela que l'utilisation d'un additionneur dans l'architecture pipeline, permet d'obtenir un compromis entre la surface d'implantation et la rapidité.

### 3.2.4. Additionneur pipeline proposé pour le terme $(\omega t + \varphi_k)$

Cet additionneur utilisé dans l'architecture universelle de la figure IV.3, calcule la somme de la phase « $\varphi_k$ » (correspondant au symbole transmis $k$) et le terme produit par l'accumulateur de phase, donnant ainsi le résultat sous le terme : $\omega t + \varphi_k$. Plus précisément, à chaque cycle d'horloge, l'additionneur fait l'opération d'addition de la phase $\varphi_k$ avec un nouveau terme produit par l'accumulateur de phase ($F_{cw}$), jusqu'à la formation d'une période du signal à transmettre.

Nous avons ainsi, proposé un additionneur rapide de structure pipeline adapté à cette opération d'addition. Leur architecture interne est composée de trois additionneurs et de registres.

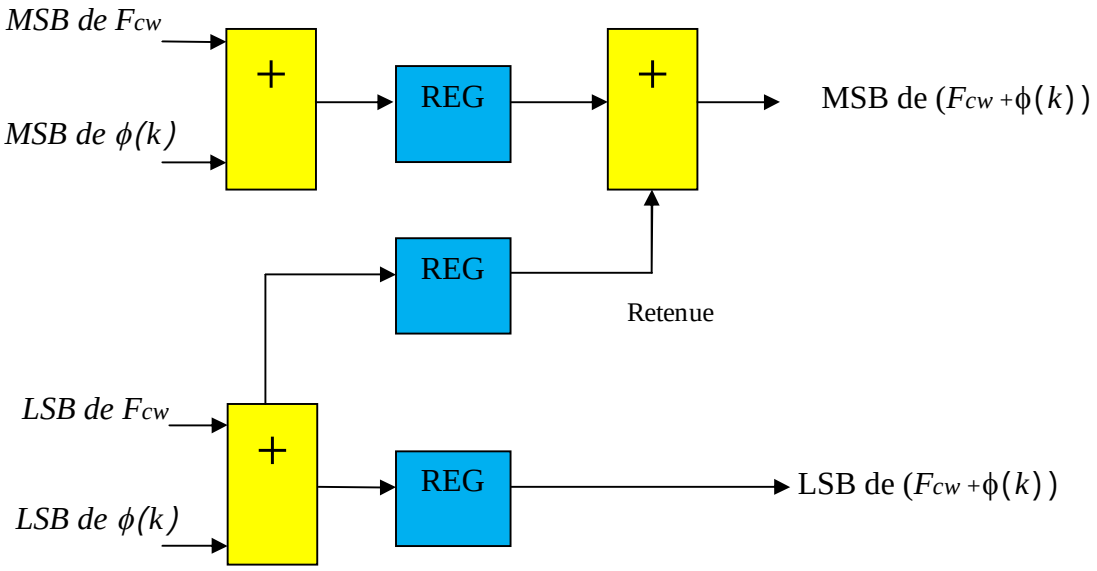


Figure IV.6. L'architecture pipeline de l'additionneur.

Dans notre cas, les deux entrées sont la phase et le terme $F_{cw}$. Un additionneur pour le calcul de la somme du point fort (MSB) pour les deux entrées, un additionneur pour le calcul de la somme du point faible (LSB) pour les deux entrées, et le troisième additionneur calcule la somme du résultat du premier additionneur (MSB) et la retenue (figure IV.6).

### 3.2.5. Générateurs de forme d'onde (Générateurs sinusoïdes)

Notre architecture proposée est basée sur l'équation 4, il reste donc à générer le cosinus du terme $\omega t + \varphi_k$, puis multiplier le cosinus avec l'amplitude $A_k$ afin d'obtenir un signal qui représente la modulation M-QAM. Dans le modulateur QAM (figure IV.1), les valeurs des cosinus et sinus sont stockées dans des mémoires (ou LUTs). Cette solution nécessite de grandes mémoires mortes si on cherche à augmenter la précision. Cette solution ne peut être parallélisée que par le dédoublement des mémoires ou l'utilisation de mémoires à accès multiples simultanés [93]. Afin d'éviter l'utilisation des mémoires, nous avons proposé l'algorithme CORDIC pour le calcul du cosinus du terme $\omega t + \varphi_k$ (l'algorithme Cordic est détaillé précédemment). Le CORDIC permet de faire le calcul des fonctions trigonométriques comme le cosinus et le sinus par approximation successives [4][92]. Par conséquent, le terme $\omega t + \varphi_k$ avec le terme $A_k$ représentent les deux entrées du module Cordic. Le principe du Cordic est simple [92][93]: Soit un vecteur $\vec{v}$, et un angle θ tels que:

$$\vec{v} = \begin{pmatrix} A_k \\ 0 \end{pmatrix} \quad (6)$$

$$\theta = \omega t + \varphi_k \quad (7)$$

L'algorithme produit une rotation qui conduit au vecteur résultant:

$$\vec{v}' = \begin{pmatrix} A_k\, cos(\omega t + \varphi_k) \\ A_k\, sin(\omega t + \varphi_k) \end{pmatrix} \quad (8)$$

Nous avons utilisé l'architecture du processeur CORDIC pipeline optimisé [93] qui permet de minimiser la consommation en ressource dans le FPGA.

### 3.2.6. Principe de fonctionnement de l'architecture proposée

Le principe de fonctionnement de cette architecture est simple car toutes les données, quelque soit le type de modulation QAM choisi (4-QAM, 16-QAM,…, où 1024-QAM), seront représentées dans un format compatible avec la modulation QAM 1024. Cette tâche est réalisée par le codeur, puis les 10 bits sortant du codeur sont à la fois l'entrée d'une mémoire (LUT) contenant la phase ($\phi_k$) correspondant à chaque symbole et l'entrée d'une autre LUT contenant l'amplitude ($A_k$) correspondant à chaque symbole. En même temps, l'accumulateur de phase, assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. La phase générée par la mémoire (LUT) est additionnée avec la sortie de l'accumulateur de phase en utilisant l'architecture de l'additionneur pipeline proposée. Enfin, le résultat de l'addition de $(\omega t + \phi_k)$ et de l'amplitude $(A_k)$ est réalisé par le processus Cordic. Ce dernier calcule le cosinus de $(\omega t + \phi_k)$, puis il génère le signal QAM "$A_k.cos(\omega t + \phi_k)$" à transmettre.

## 3.3. Parallélisation de l'architecture proposée pour la modulation M-QAM

Pour produire '*m*' fois plus d'échantillons d'une même sinusoïde, on parallélise '*m*' processeurs CORDIC, où chaque processeur génère un échantillon comme représenté dans la figure (IV.7). En effet, plusieurs processeurs CORDIC parallélisés en '*m*' étages sont capables de produire '*m*' fois plus d'échantillons d'une même sinusoïde d'un seul processeur CORDIC.

Cette architecture parallèle augmente la fréquence de fonctionnement malgré l'augmentation de la consommation de ressource en FPGA. Le sérialiseur rapide dans la figure (IV.7) est utilisé pour reconstituer successivement les échantillons produits par chacun des étages afin de générer le signal de sortie.

L'architecture nécessite aussi la parallélisation de plusieurs étages d'accumulateurs de phase et d'additionneurs au même nombre que celui des étages de CORDIC. On peut aussi supprimer les étages d'accumulateurs de phase et lire directement les valeurs « *m* *$F_{cw}$ » depuis une table d'allocation (figure IV.7). Ceci permettra de réduire la consommation de surface et de ressource de cette architecture parallèle. On peut aussi imaginer que l'utilisation du processeur Cordic optimisé de [93] va réduire la consommation en ressource mieux que si on utilise un processeur Cordic classique.

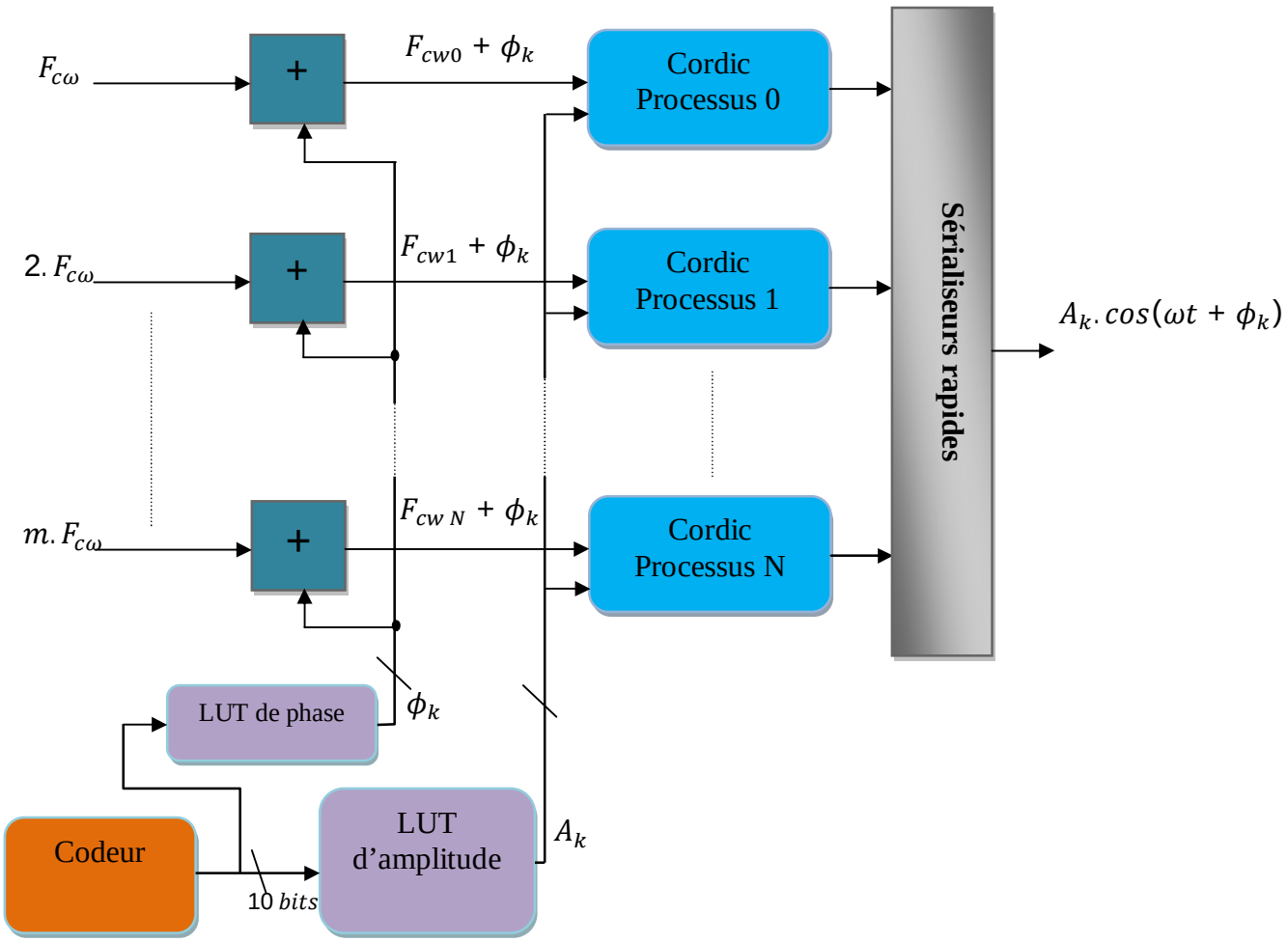


Figure IV.7. Parallélisation de l'architecture proposée pour la modulation M-QAM.

### 3.4. Architecture générique pour la Modulation QAM, PSK et FSK

Sur la base de l'architecture de modulation QAM numérique développée dans la section précédente, nous avons proposé une nouvelle architecture générique à usage universel (figure IV.8), qui peut prendre en charge les modulations M-QAM, M-PSK et B-FSK. Les équations des signaux modulés en phase (PSK), la modulation QAM, et la modulation FSK sont ainsi décrites [9] [112] :

- Pour la Modulation par Déplacement de Phase (MDP) qui est aussi appelée par son abréviation anglaise: PSK pour "Phase Shift Keying", l'amplitude et la fréquence sont fixes par contre la phase change avec le symbole à transmettre. On pourrait imaginer que les symboles seraient disposés de façon quelconque sur un cercle. L'équation qui décrit un signal modulé en phase (PSK) est donnée par :

$$s(t)_{PSK} = A. cos(\omega t + \theta_k) \qquad (9)$$

- Pour la modulation QAM, nous avons décrit son principe de fonctionnement, et on donne l'équation qui décrit un signal modulé en modulation QAM:

$$s(t)_{QAM} = A_k.cos(\omega t + \phi_k) \quad (10)$$

- Pour la Modulation par Déplacement de Fréquence (MDF) qui est aussi appelée par son abréviation anglaise: FSK pour "Frequency Shift Keying", l'amplitude est fixe par contre c'est la fréquence qui change avec le symbole à transmettre.

Nous supposons que la modulation FSK traitée par l'architecture proposée est la modulation FSK binaire c'est-à-dire M = 2 (1 ou 0). Dans ce cas, le signal modulé peut être écrit très simplement comme étant composé de deux porteuses différentes.

$$\begin{cases} s_1(t)_{FSK} = A.\cos(2\pi f_1 t) \\ s_2(t)_{FSK} = A.\cos(2\pi f_2 t) \end{cases} \quad (11)$$

$s_1(t)_{FSK}$ la réponse d'un bit égal à 1, et $s_2(t)_{FSK}$ la réponse d'un bit égal à 0. Ceci peut être aussi écrit comme une déviation ($\Delta f$) par rapport à la fréquence porteuse.

$$\begin{cases} s_1(t)_{FSK} = A.\cos\left(2\pi\left(f - \frac{\Delta f}{2}\right)t\right) = A.\cos(2\pi(\Delta f_1)t) \\ s_2(t)_{FSK} = A.\cos\left(2\pi\left(f + \frac{\Delta f}{2}\right)t\right) = A.\cos(2\pi(\Delta f_2)t) \end{cases} \quad (12)$$

Par conséquent, l'équation qui décrit un signal modulé en fréquence (FSK) est donnée par :

$$s(t)_{FSK} = A.\cos\left(\omega t + \frac{2\pi\Delta f}{2} a_k t\right) \quad (13)$$

L'architecture proposée représentée par la figure IV.8 est composée de:

- une LUT qui est utilisée pour la modulation QAM et également pour la modulation PSK, car la modulation de phase PSK peut être considérée comme une modulation MAQ où l'amplitude est constante. Donc la LUT (LUT-QAM-PSK dans la figure IV.8) contient les phases d'un signal modulé selon les modulations QAM et PSK,
- un processus CORDIC chargé de générer le signal à transmettre,
- un accumulateur de phase assure la génération des *n* points nécessaires à la description discrète du signal sinusoïdal désiré,
- un codeur utilisé pour la modulation QAM et la modulation PSK,
- Un bloc LUT d'amplitude, qui est utilisé seulement pour la modulation QAM puisque dans le cas de la modulation de phase et de la modulation de fréquence, l'amplitude est fixe. Ce bloc génère donc l'amplitude $A_k$.
- Un multiplexeur (couleur noir) est utilisé seulement pour la modulation B-FSK commandé par le symbole $a_k$ (ce symbole peut prendre deux valeurs '0' ou '1').
- Trois commandes C1, C2 et C3 (figure IV.8) nous permettent de changer le mode de fonctionnement (type de modulation) à travers trois multiplexeurs de couleur grise dans la figure (IV.8).

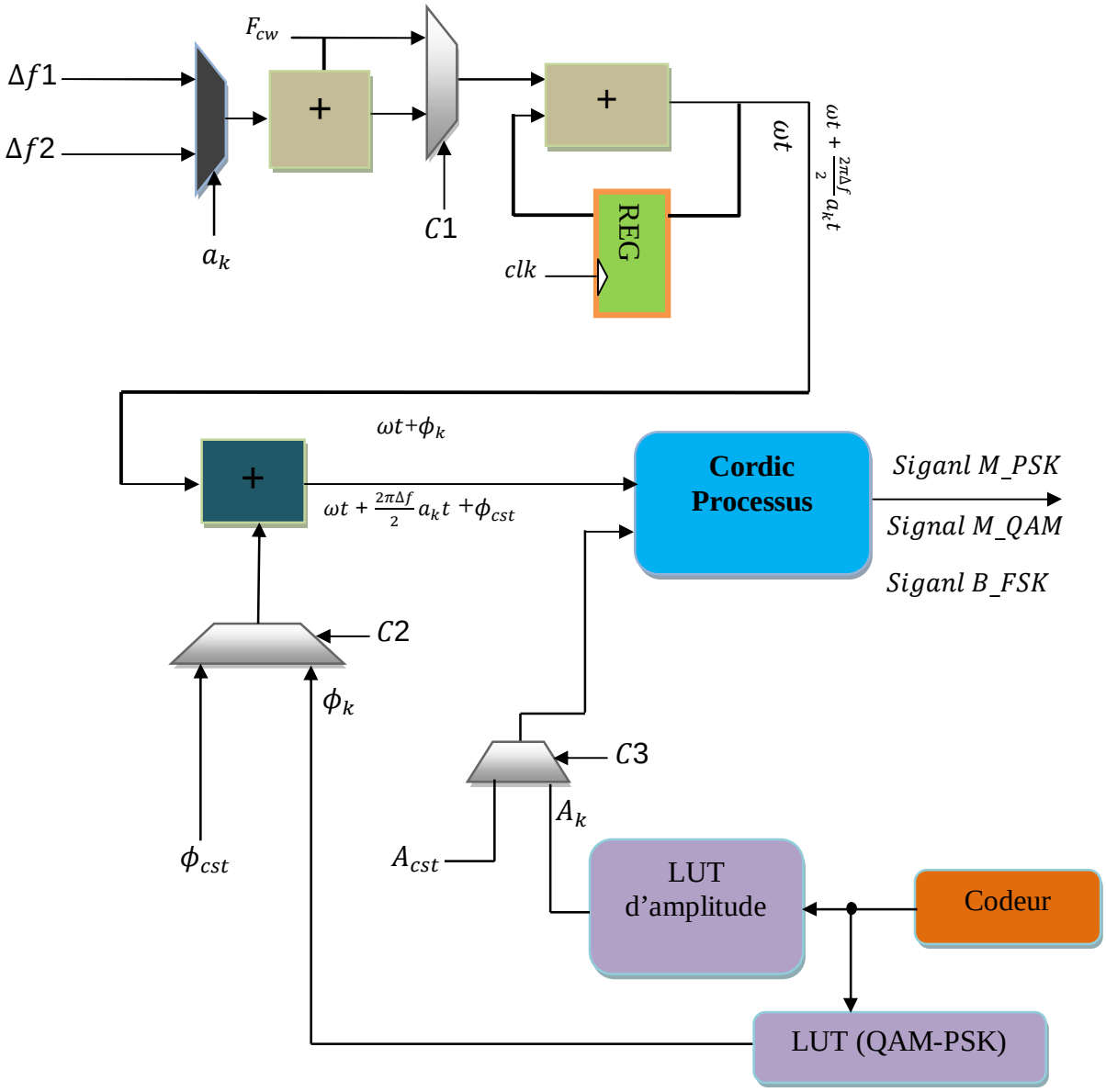


Figure IV.8. Architecture universelle pour la modulation M-QAM, M-PSK et B-FSK.

Le principe de fonctionnement est décrit par les trois modes suivants:

- **Mode 1: Modulation M-QAM**

Dans ce cas deux paramètres sont variables en fonction du symbole transmis; la phase et l'amplitude, par contre la fréquence reste fixe. Le codeur génère une trame de 10 bits pour la modulation M-QAM, où M peut varier de 2 jusqu'à 1024. La sortie de ce codeur représente une adresse de lecture pour la mémoire (LUT-QAM-PSK) qui génère la phase correspondante à chaque symbole. Selon cette adresse, le bloc LUT d'amplitude génère l'amplitude ($A_k$) correspondante à ce symbole. Dans ce mode, les paramètres de reconfiguration C1, C2, et C3 dans la figure (IV.8) permettent de sélectionner le mot

« $F_{cw}$ », la phase "$\phi_k$" et l'amplitude "$A_k$" respectivement. Simultanément, l'accumulateur de phase assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. Dans ce cas, l'accumulateur de phase est piloté par le mot de contrôle de la fréquence « $F_{cw}$ » correspondant à l'incrément de phase appliqué à chaque période d'horloge du système. La phase générée par la mémoire (LUT) est additionnée avec la sortie de l'accumulateur de phase. Enfin, la sortie de l'additionneur $(\omega t + \phi_k)$ et l'amplitude $(A_k)$ sont utilisées par le processus Cordic. Ce dernier calcule le cosinus de $(\omega t + \phi_k)$, puis il génère le signal M-QAM $(A_k.cos(\omega t + \phi_k))$ à transmettre.

- **Mode 2: Modulation M-PSK**

Dans ce cas, la phase $\phi_k$ est variable en fonction du symbole transmis; donc l'amplitude et la fréquence sont fixes. Le codeur génère une trame de 10 bits pour la modulation PSK. Cette trame de 10 bits représente une adresse de lecture pour la mémoire (LUT-QAM-PSK) contenant la phase correspondante de chaque symbole. Dans ce mode, les paramètres de reconfiguration C1, C2, et C3 dans la figure (IV.8) permettent de sélectionner le mot « $F_{cw}$ », la phase "$\phi_k$" et l'amplitude constant "$A_{cst}$" respectivement. Simultanément, l'accumulateur de phase assure la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. La phase générée par la mémoire (LUT-QAM-PSK) est additionnée avec la sortie de l'accumulateur de phase. Enfin, la somme $(\omega t + \phi_k)$ et la valeur $A_{cst}$ représentent les deux entrées nécessaires pour le processus Cordic qui va générer le signal à transmettre modulé en phase (M-PSK).

- **Mode 3 : Modulation B-FSK**

Ce mode utilise le changement de la fréquence en fonction du symbole transmis $a_k$. Dans notre architecture, la valeur du symbole à transmettre $a_k$ (0 ou 1), permet de sélectionner la valeur de la déviation qui correspond à la modulation de fréquence (équations 12 et 13) en utilisant un multiplexeur (coloré en noir dans la figure IV.8). Simultanément, la valeur de la déviation correspondant au symbole $a_k$ est ajoutée au mot de contrôle de la fréquence « $F_{cw}$ » correspondant à l'incrément de phase appliqué à chaque période d'horloge du système. Ainsi, le paramètre de configuration C1 permet de sélectionner le résultat de l'addition de $(F_{cw} + a_k.\Delta f)$. Dans ce cas, l'accumulateur de phase est piloté par le mot de contrôle de la fréquence « $F_{cw} + a_k.\Delta f$», il assure donc avec ce nouveau mot de contrôle, la génération des N points nécessaires à la description discrète du signal sinusoïdal désiré. Les paramètres de reconfiguration C2, et C3 dans la figure (IV.8) permettent de sélectionner respectivement une phase constant "$\phi_{cst}$" et une amplitude constant "$A_{cst}$". Enfin, la sortie de l'accumulateur de phase et la valeur de l'amplitude représentent les deux entrées nécessaires pour le processus Cordic qui va générer et transmettre le signal modulé en fréquence (B-FSK).

Cette architecture générique pour la modulation M-QAM, M-PSK et B-FSK peut être parallélisée pour augmenter la fréquence de fonctionnement et générer plus d'échantillons.

## 4. Architecture générique pour la démodulation numérique et le codage.

Dans cette partie, nous proposons une nouvelle architecture universelle utilisée pour quatre fonctionnalités. Elle peut être utilisée comme un démodulateur de phase (PSK), comme un démodulateur de fréquence (FSK), comme un générateur de code de brouillage connu sous le nom de code Gold utilisé dans les systèmes WCDMA, et comme un code convolutif (codage de canal). Cette architecture est basée sur un simple registre LFSR (*Linear Feedback Shift Register*) configurable. L'idée principale est de proposer une structure générique qui permet de basculer entre les quatre fonctions par un simple changement de paramètres, en suivant la technique de paramétrisation et l'approche d'opérateurs communs.

### 4.1.Registres à décalage à rétroaction linéaire (LFSR)

Lorsqu'une étude bibliographique est faite sur les architectures numériques pour la modulation, le codage de canal, le filtrage etc., nous pouvons trouver de nombreux points de similarités entre ces architectures [2] [4] [113] [114]. Parmi les points de similarités intéressants, on remarque que les registres LFSR sous forme binaire, sont presque présents dans tous les systèmes électroniques actuels. Ils jouent un rôle essentiel dans la génération de séquences, le codage de canal (correction et détection d'erreurs), les codes de brouillages et le filtrage [2][115]. Nous pouvons les trouver dans les téléphones mobiles, les modems et presque toutes les applications de codage [2].

Un registre à décalage à rétroaction linéaire, ou LFSR (acronyme de l'anglais *linear feedback shift register*) est un registre où lorsque l'horloge avance (l'horloge du circuit où l'architecture est implantée), le signal se déplace à travers le registre, d'une bascule à une autre (figure IV.9). Un LFSR peut être formé en réalisant un **OU** exclusif sur les sorties des deux ou plusieurs bascules ($r_i$ sur la figure IV.9) qui alimentent de nouveau l'entrée de l'une des bascules (figure IV.9). Comme le registre possède un nombre fini d'états, il doit entrer dans un cycle répétitif. Cependant, un LFSR avec une fonction de rétroaction bien choisie peut produire une séquence de bits qui apparaît dans la nature aléatoire et qui a un très long cycle [93] [116].

Les deux types d'architectures (Many-To-One et One-To-many) permettant de réaliser des LFSR sont présentés sur la figure (IV.9). Un choix judicieux des coefficients $\{c_i\}_{1\leq i<n}$ et $\{c'_i\}_{1\leq i<n}$ fournit l'état de ces deux architectures. En outre, les coefficients "$c_i$" peuvent prendre deux valeurs "1" ou "0". Ainsi, quand " $c_i$ = 1" il y a une connexion physique, et lorsque " $c_i$ = 0", il n'y a pas de connexion physique.