

3/02

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Mémoire de fin d'études

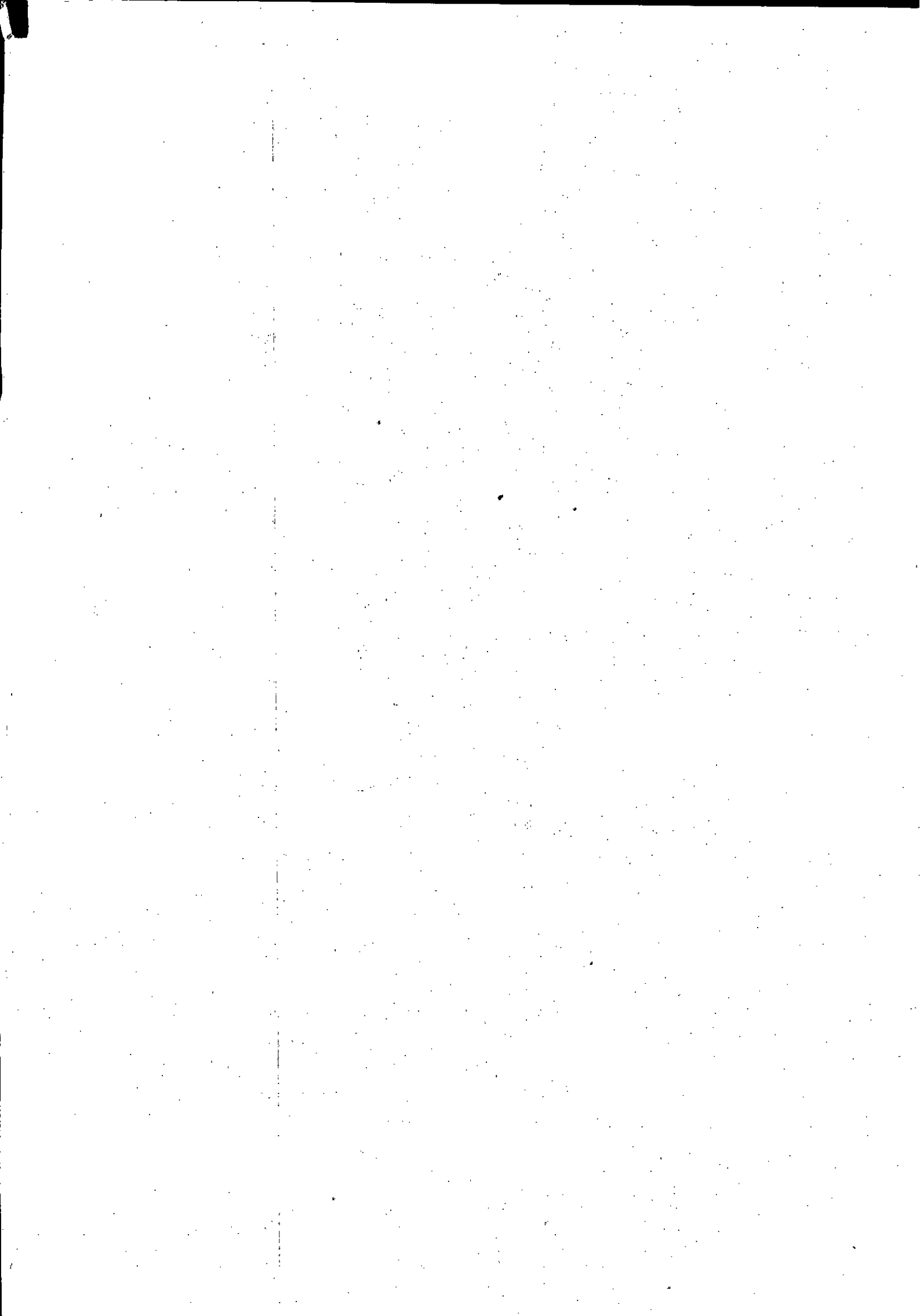
En vue de l'obtention du diplôme d'ingénieur d'Etat en
Electronique

**Etude et implémentation du codeur
LPC sur une carte DSP TMS320C31**

Dirigé par :
M^r D.BERKANI
M^r R.SAADOUN

Réalisé par :
M^r BOUAFIA Khaled

Promotion : Juin 2002



REMERCIEMENTS

Ce travail a été effectué au laboratoire "Signal & Communications" du département d'Electronique à l'Ecole Nationale Polytechnique (ENP).

Mes vifs remerciements s'adressent à mon encadreur Monsieur D.BERKANI, Professeur à l'ENP et responsable du laboratoire "Signal & Communications", pour ses conseils, encouragements, et aides précieux qu'il m'a apporté pendant la réalisation de mon travail.

Je remercie Monsieur R.SADOUN, chargé de cours à l'ENP, pour ses conseils qu'il m'a apporté pendant mon travail.

Je remercie Monsieur R.AKSAS, professeur à l'ENP, pour l'honneur qu'il m'a fait en acceptant de présider le jury.

Je remercie Monsieur B.BOUSSEKSSOU, chargé de cours à l'ENP, qui m'a fait l'honneur de faire partie du jury de soutenance.

Je tient à exprimer mes remerciements à M^{me} HAMAMI et M^{lle} MERAZKA, pour leurs aides qu'elles m'ont apporté pendant mon travail.

ESTABLISHED 1880
SOUTH BEND, INDIANA
S. B. BROS. & CO. INC.

ملاحظتان :

- الهدف من هذا العمل هو دراسة وادماج المشفر LPC على لوحة
TMS320C31 DSK. عملية التشفير تتمثل في ثلاث مراحل
- التحليل : أي تحليل إشارة الكلام بواسطة التحويلات الخطية
 - تكميم الودائط أي رسايط إشارة الكلام.
 - إعادة تركيب إشارة الكلام باستعمال رسايطها.
- المعانيخ : DSK : لوحة DSP اللطوبين ، DSP : معالسون معالجة الإشارة
LPC : مشفر تنبئي خطي .

Abstract:

The aim of this work is the studies and implementation of the LPC speech voice coder on the TMS320C31 DSK. The speech coding consist of tree stages:

- Analysis of signal by the linear prediction.
- Quantization of parameters of signal.
- Synthesis of signal.

Keywords: DSK : DSP Starter Kit , DSP : Digital Signal Processor,
LPC: Linear Predictive Coding

Résumé :

Le but de ce travail est l'étude et l'implémentation du codeur LPC sue le TMS320C31 DSK. L'opération du codage consiste en trois étapes :

- Analyse du signal vocal par la prédiction linéaire.
- Quantification des paramètres du signal vocal.
- Reconstitution du signal vocal.

Mot clés : DSK : Kit de Développement DSP, DSP : processeur de traitement du signal, LPC :Codeur Prédicatif Linéaire.

SOMMAIRE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Remerciements	
Dédicaces	
Résumé	
Introduction générale.....	4
I) Généralités.....	5
1) Introduction.....	6
2) Mécanisme de la phonation.....	6
3) Propriétés statistiques du signal vocal.....	8
4) Systèmes linéaires invariants.....	9
5) Modèle de production de la parole.....	10
6) Conclusion.....	11
II) Codeur prédictif linéaire.....	12
1) Introduction.....	13
2) Analyse par la prédiction linéaire.....	13
2.1) Méthode d'autocorrélation.....	13
2.2) Méthode de covariance.....	14
2.3) Algorithme de résolution.....	15
Algorithme de Wiener-Levinson-Durbin.....	16
2.4) La pré-accentuation.....	16
2.5) Considérations pratiques.....	17
3) Analyse du signal vocal.....	18
3.1) Les coefficients a_i	18
• Calcul des LSF.....	18
3.2) Le signal erreur (résidu).....	19
4) Reconstitution du signal vocal.....	20
4.1) Calcul des coefficients a_i	20
5) Codage et Décodage des paramètres LSF_i	20
6) modèle du codeur LPC.....	20
7) Conclusion.....	21
III) Etude du DSP TMS320C31.....	22
1) Introduction.....	24
2) Classes de signaux à traiter.....	24
3) Domaines d'application des DSP.....	24
3.1) Domaine des télécommunications.....	25
3.2) Domaine militaire.....	25
3.3) Domaine des multimédia et du grand public.....	25
3.4) Domaine médical.....	25
3.5) Domaine de l'automatique.....	25
3.6) Domaine de l'instrumentation.....	25
4) Architecture.....	25
4.1) Description générale.....	25
4.2) Présentation matérielle.....	25
4.3) Description et rôle des signaux.....	26



4.4) Diagramme bloc du C31.....	28
4.4.1) Le Multiplieur.....	28
4.4.2) L'unité arithmétique et logique (UAL)	28
4.4.3) L'unité de calcul des registres Auxiliaires (ARAU)	29
4.4.5) Les Registres.....	29
• Les registres de précision étendue (R0-R7).....	29
• Les registres auxiliaires (AR0-AR7)	29
• Les registres d'index (IR0-IR1)	29
• Le pointeur de pile système (SP)	29
• Le registre d'état (ST)	29
• Le registre d'interruption CPU/DMA (IE)	30
• Le registre indicateur d'interruption CPU (IF)	31
• Le registre indicateur d'E/S (IOF)	31
• Les registres de répétition de bloc (RC-RS-RE)	32
• Le compteur programme (PC)	32
• La mémoire interne.....	32
4.5) Les périphériques.....	32
4.5.1) Le timer.....	32
• Le registre de contrôle globale (GCR)	33
• Le registre de période.....	34
• Le registre du comptage.....	34
4.5.2) Le port série.....	34
• Le registre de contrôle globale (GCR)	35
• Le registre de contrôle des broche FSX/DX/CLKX.....	37
• Le registre de contrôle des broche FSR/DR/CLKR.....	37
• Le registre de contrôle du timer de réception/transmission.....	39
• Le registre de comptage du timer transmission/réception.....	40
• Le registre période du timer transmission/réception.....	40
• Le registre de transmission de données.....	40
• Le registre de réception de données.....	41
• La carte mémoire du DSP TMS320C31.....	42
• La carte mémoire des périphériques.....	43
5)Les modes d'adressage.....	43
5.1) Adressage par registres.....	44
5.2) Adressage direct.....	44
5.3) Adressage immédiat.....	44
5.4) Adressage relatif PC.....	44
5.5) Adressage indirect.....	44
5.5.1) Adressage indirect avec déplacement.....	44
5.5.2) Adressage indirect avec registres d'index.....	45
5.5.3) Adressage indirect (cas spéciaux).....	45
5.6) Adressage circulaire.....	45
5.6) Adressage par inversion de bits.....	46
6) Conclusion.....	47

IV) Etude du TMS320C31 DSP Starter Kit.....	48
1) Introduction.....	49
2) Description générale.....	49
3) Diagramme bloc du TMS320C31 DSK.....	49
4) L'interface analogique AIC (Analogue Interface Controller)	49
4.1) Diagramme bloc de l'AIC.....	50
4.2) Brochage du circuit.....	51
4.3) Utilisation du TLC32040.....	52
a) initialisation de l'AIC.....	52
b) initialisation du Timer.....	52
c) initialisation du Port Série.....	52
d) programmation de l'AIC.....	53
• Communications primaires.....	53
• Communications secondaires.....	54
• signification des bits du registre de contrôle.....	54
5) Conclusion.....	55
V) Mise en œuvre du codeur LPC.....	56
1) introduction.....	57
2) Description du programme.....	57
2.1) L'initialisation des périphériques.....	57
2.2) Les entrées/sorties.....	57
2.3) Analyse du signal vocal.....	57
2.4) Codage et décodage des paramètres LSF_i	57
2.5) synthèse du signal vocal.....	58
2.6) Evaluation du codeur.....	59
3) Conclusion.....	59
Conclusions générales.....	60
Bibliographie.....	61
Annexe.....	63

Introduction

La réduction du débit binaire pour la transmission et l'enregistrement du signal vocal ont toujours suscité un très grand intérêt et des algorithmes de plus en plus complexes peuvent être exécutés en temps réel [1].

Le système utilisé couramment pour la téléphonie digitale est le système PCM (Pulse Code Modulation ou modulation par impulsions codées). Chaque échantillon du signal est alors représenté et codé indépendamment.

Une réduction du débit binaire peut être obtenue si on spécule sur la corrélation qui existe entre les échantillons successifs du signal vocal. On peut donc quantifier et transmettre la différence entre deux échantillons successifs en lieu et place des deux échantillons. Cette méthode est appelée MIC Différentiel (DPCM, Differential Pulse Code Modulation).

Une amélioration sensible des performances du système MIC Différentiel peut être obtenue si le pas de quantification est adapté à l'application visée (ADPCM, Adaptive Differential Pulse Code Modulation). Le débit obtenu avec cette méthode est de 32 kbits/s et ses principaux avantages sont une relative simplicité d'implémentation et de courts délais de calcul. Elle est ainsi utilisée dans un nombre élevé d'applications, telles qu'une transmission pour les réseaux sans fils (*wireless communications*) et les communications par satellite.

Pour atteindre des débits plus faibles, l'introduction du modèle de reproduction de la parole apporte une nette amélioration [4]. Les paramètres du modèle sont quantifiés au lieu signal vocal. De nombreux algorithmes de codages paramétriques ont été proposés, parmi lesquels :

- Linear Predictive Coding (L.P.C).
- Code Exited Linear Prediction (C.E.L.P).
- Multi Pulse Linear Predictive Coding (M.P.L.P.C).

Dans ce mémoire, le but de notre travail est l'implémentation du codeur LPC dans une carte DSP TMS320C31. Ce mémoire comporte cinq chapitres :

Le premier chapitre est consacré à l'étude du signal vocal et ses propriétés statistiques. Les systèmes linéaires et stationnaires ainsi que le modèle de production de la parole y sont présentés.

Le deuxième chapitre est consacré à l'analyse par prédiction linéaire et à la description du codeur L.P.C.

Le troisième chapitre est consacré à l'étude du DSP TMS320C31.

Le quatrième chapitre est consacré à la description de la carte DSP Starter Kit, sur laquelle le codeur LPC est implémenté.

Le cinquième chapitre expose la mise en œuvre du codeur LPC.

I) Généralités.....	5
1) Introduction.....	6
2) Mécanisme de la phonation.....	6
3) Propriétés statistiques du signal vocal.....	8
4) Systèmes linéaires invariants.....	9
5) Modèle de production de la parole.....	10
6) Conclusion.....	11

1) Introduction

Ce chapitre comporte des généralités regroupant des notions de production et les propriétés statistiques du signal vocal. Nous donnons un aperçu sur les systèmes invariants et leurs utilisations dans la modélisation de la production de la parole.

2) Mécanisme de la phonation

La parole peut être décrite comme le résultat de l'action volontaire et coordonnée d'un certain nombre de muscles. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par les sensations cénesthésiques.[3]

L'appareil respiratoire fournit l'énergie nécessaire lorsque l'air est expiré par la trachée-artère. Au sommet de celle-ci se trouve le larynx, dont le rôle est de moduler la pression de l'air qui est ensuite appliqué au conduit vocal qui s'étend du pharynx jusqu'aux lèvres.

Le larynx est un ensemble de muscles et de cartilages mobiles qui entourent une cavité située à la partie supérieure de la trachée (Fig. 1.1).

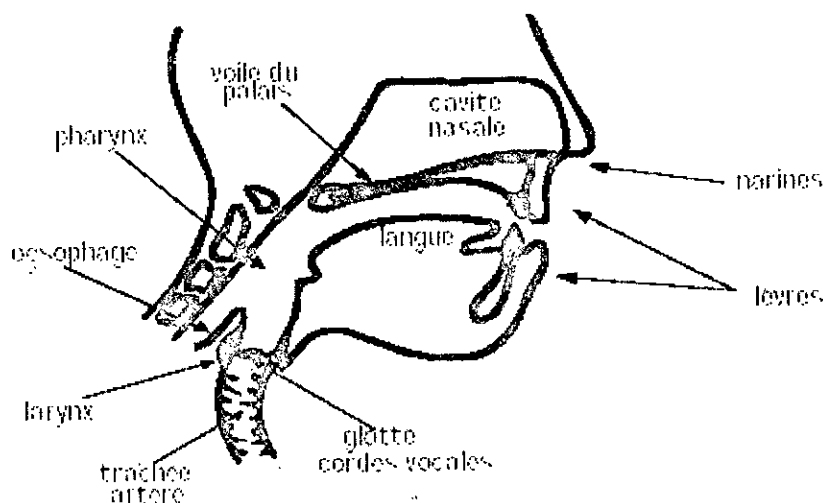


Figure 1.1 L'appareil phonatoire humain [3]

Les cordes vocales sont en fait deux lèvres symétriques placées en travers du larynx ; ces lèvres peuvent fermer complètement le larynx et, en s'écartant progressivement, déterminer une ouverture triangulaire appelée glotte (fig 1.2). L'air y passe librement pendant la respiration, la voix chuchotée, la phonation des sons sourds et non voisés, c'est-à-dire dont la structure n'est pas périodique.

Les sons voisés résultent au contraire d'une vibration périodique des cordes vocales. Le conduit vocal, qui est traversé par le flot de l'air, peut prendre des formes variables de par les positions diverses des lèvres, de la langue, des mâchoires et du voile du palais.

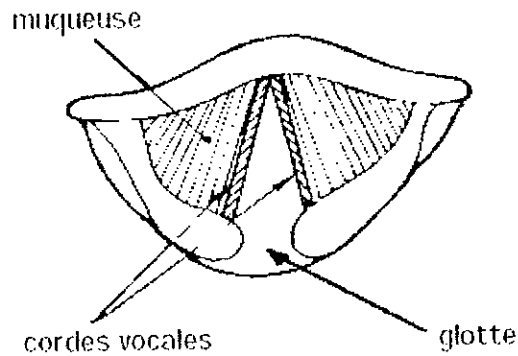


Figure 1.2 vue de haut du larynx [1]

On distingue deux types majeurs de sons émis par l'être humain :

- Les sons voisés, comme les voyelles, dont la fréquence fondamentale (pitch) peut varier :
 - De 80 à 200 Hz pour une voie masculine.
 - De 150 à 450 Hz pour une voie féminine.
 - De 200 à 600 Hz pour une voie d'enfants.

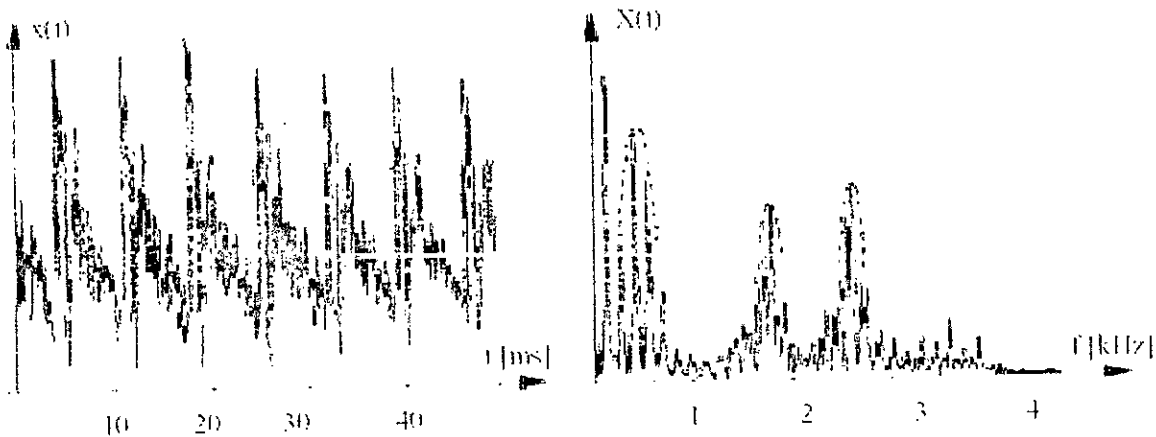


Figure (1.3) un signal voisé et son spectre [1]

- Les sons non voisés, comme certaines consonnes, ne présentent pas une structure périodique, ils peuvent être considérés comme un bruit blanc filtré par la transmittance de conduit vocal (fig 1.4). Le spectre d'un son non voisé ne présente pas de structure de pitch.

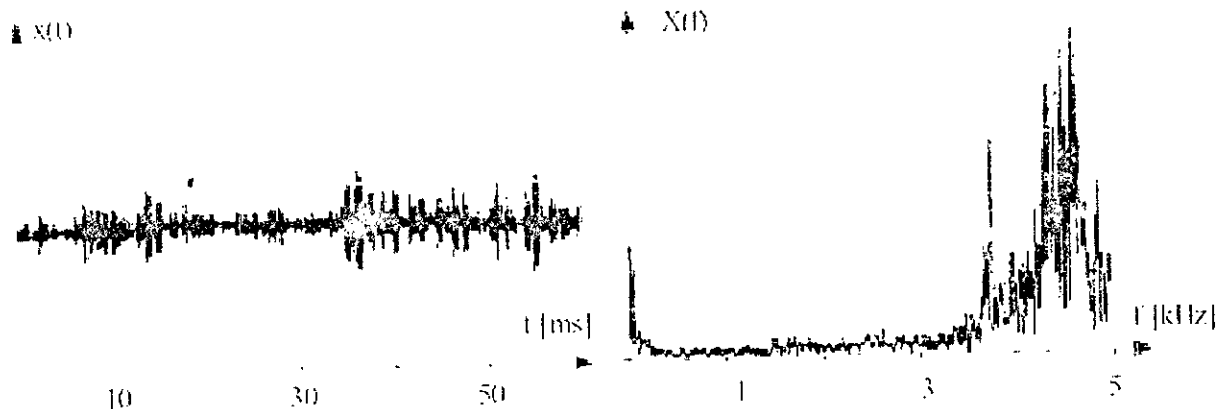


Figure (1.4) un signal non voisé et son spectre [1]

3) Propriétés statistiques du signal vocal

Un signal vocal est une réalisation particulière un processus aléatoire et non stationnaire, ceci implique que ses propriétés statistiques doivent être estimées sur des intervalles de temps importants (plusieurs dizaines de secondes) et moyennées pour plusieurs locuteurs [1]. Ces statistiques sont appelées "à long terme".

En principe le concept de densité spectrale ne s'applique qu'à un signal stationnaire. Cependant, le signal vocal normal est essentiellement non stationnaire. C'est pourquoi on définit la notion de propriétés statistiques « à court terme » qui sont estimées sur des tranches temporelles d'une durée de 10 à 30 ms pendant lesquelles le signal est quasiment stationnaire.

La densité de probabilité à long terme pour le signal vocal présente une valeur importante lorsque l'amplitude des échantillons est nulle. Ceci traduit le fait qu'il existe beaucoup d'intervalles de silence dans une conversation, même continue.

La fonction d'autocorrélation d'un signal stationnaire est définie par:

$$R_{xx}(k) = \lim_{N \rightarrow \infty} \left[\frac{1}{2N+1} \sum_{n=-N}^{+N} x(n) \cdot x(n+k) \right] \quad (1.1)$$

N : nombre d'échantillons.

$x(n)$: amplitude du $n^{\text{ème}}$ échantillon.

On définit ensuite la densité spectrale de puissance $S_{xx}(\omega T_e)$ comme la transformée de Fourier de la fonction d'autocorrélation.

$$S_{xx}(\omega T_e) = \sum_k R_{xx}(k) \cdot \exp(-jk\omega T_e) \quad (1.2)$$

$R_{xx}(k)$: fonction d'autocorrélation.

T_e : période d'échantillonnage.

4) Systèmes linéaires invariants

Un système de traitement numérique agit sur un signal numérique d'entrée, appelé excitation $u(n)$, et produit un autre signal numérique à sa sortie, appelé réponse $s(n)$.

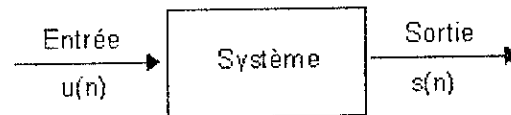


Figure 1.5 système de traitement numérique

Dans la majorité des cas rencontrés en pratique, les systèmes peuvent être considérés linéaires et invariants dans le temps.

Les termes linéaires et invariants indiquent respectivement que la propriété de linéarité est vérifiée (une combinaison linéaire de deux signaux d'entrées implique la même combinaison pour leur réponse) et que les caractéristiques temporelles des signaux telles que les retards ne changent pas à cause du système.

On appelle équation récurrente linéaire à coefficients constants une expression de la forme :

$$s(n) + \sum_{k=1}^p a_k s(n-k) = G \sum_{l=0}^q b_l s(n-l), \quad b_0=1 \quad (1.3)$$

où $\{a_k\}$, $\{b_l\}$ et le gain G sont les paramètres du système.

Dans le domaine fréquentiel, la fonction de transfert en Z de l'équation récurrente est de la forme :

$$H(z) = \frac{B(Z)}{A(Z)} = \frac{G \left[1 + \sum_{l=1}^q b_l Z^{-l} \right]}{1 + \sum_{k=1}^p a_k Z^{-k}} \quad (1.4)$$

Les racines du numérateur et du dénominateur sont, respectivement, les zéros et les pôles du système ou du modèle pôle-zéro $H(z)$.

On distingue deux cas spéciaux :

- 1) modèle tout zéro ou modèle à moyenne ajustée (MA) : $a_k=0$, $1 \leq k \leq p$.
- 2) modèle tout pôle ou modèle auto régressif (AR) : $b_l=0$, $1 \leq l \leq q$.

Le modèle pôle-zéro est appelé aussi modèle auto régressif à moyenne ajustée (ARMA).

7) Modèle de production de la parole

Un signal voisé peut être modélisé par le passage d'un train d'impulsions $u(n)$ à travers un filtre numérique récursif de type tout pôles. Cette modélisation reste valable dans le cas de sons non-voisés, à condition que $u(n)$ soit cette fois un bruit blanc. Le modèle final est illustré à la figure (1.6), et il est de la forme :

$$s(n) = - \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (1.5)$$

(où $u(n)$ est le signal d'excitation), ce qui exprime que chaque échantillon est obtenu en ajoutant un terme d'excitation à une prédiction obtenue par combinaison linéaire de p échantillons précédents. Nous verrons dans le chapitre suivant que ce modèle est adopté pour l'analyse par la prédiction linéaire

Dans le domaine fréquentiel, la fonction de transfert du modèle AR est de la forme :

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (1.6)$$

Les paramètres du modèle électrique de la phonation sont : la période du train d'impulsions (sons voisés uniquement), la décision Voisé/Non Voisé (V/NV), le gain G , et les coefficients du filtre $1/A(z)$, appelé *filtre de synthèse*.

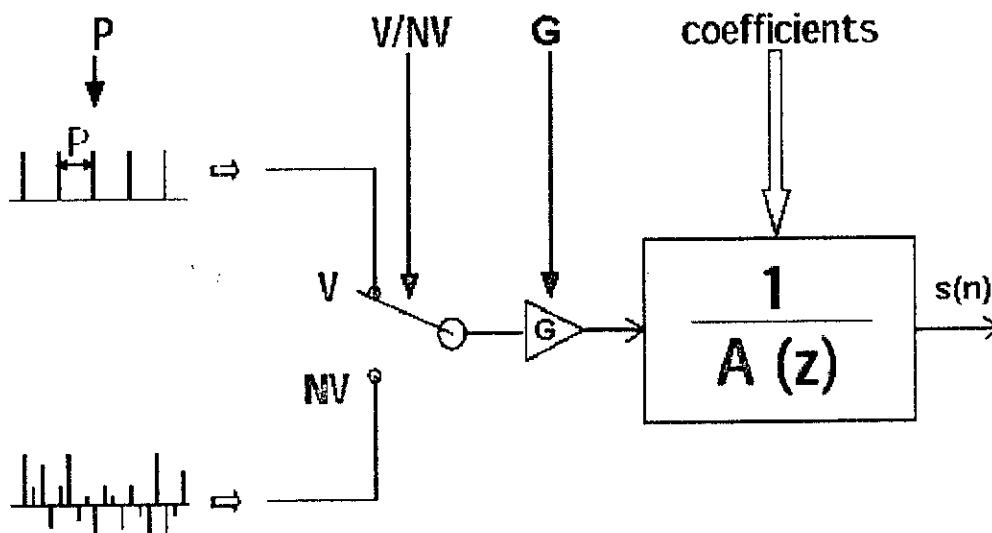


Figure 1.6 Modèle de production de la parole [3]

6) Conclusion

L'avantage de l'utilisation du modèle de production de la parole est la possibilité d'atteindre des débits plus faibles, car la redondance élevée de la forme d'onde est transformée en un ensemble de paramètres caractérisant le modèle. Une bonne modélisation nécessite une bonne connaissance des ces paramètres.

- II) Codage prédictif linéaire.....12**
- 1) Introduction.....13
- 2) Analyse par la prédiction linéaire.....13
 - 2.1) Méthode d'autocorrélation.....13
 - 2.2) Méthode de covariance.....14
 - 2.3)Algorithme de résolution.....15
 - Algorithme de Wiener-Levinson-Durbin.....16
 - 2.4)La pré-accentuation.....16
 - 2.5) Considérations pratiques.....17
- 3) Analyse du signal vocal.....18
 - 3.1)Les coefficient a_i18
 - Calcul des LSF.....18
 - 3.2) Le signal erreur (résidu).....19
- 4) Reconstitution du signal vocal.....20
 - 4.1) Calcul des coefficients a_i20
- 5) Codage et Décodage des paramètres LSF_i 20
- 6) modèle du codeur LPC.....20
- 7) Conclusion.....21

1) Introduction

Dans ce chapitre, nous exposons le principe de la prédiction linéaire qui permet d'obtenir les paramètres spectraux d'un signal vocal (court-terme). Ensuite nous ferons une description du modèle du codeur LPC.

2) Analyse par la prédiction linéaire

Cette analyse est également connue sous la dénomination d'analyse LPC (Linear Prediction Coding) et est devenue, grâce à sa grande précision et sa rapidité de calcul, la technique la plus utilisée pour l'estimation des paramètres du signal vocal.

L'idée principale de la prédiction linéaire est qu'un échantillon de parole peut être prédit comme étant égal à une combinaison linéaire d'un certain nombre, p , d'échantillons de parole le précédant immédiatement:

Ainsi le signal prédit est égale à :

$$\tilde{s}(n) = - \sum_{k=1}^p \tilde{a}_k \cdot s(n-k) \quad (2.1)$$

Les coefficients a_k sont appelés paramètres LP ou coefficients de prédiction, et le nombre p des coefficients est appelé ordre de prédiction.

L'erreur de prédiction $e(n)$ est la différence entre le signal original $s(n)$ et le signal prédit $\tilde{s}(n)$, elle est égale à :

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^p \tilde{a}_k \cdot s(n-k) \quad (2.2)$$

La transformée en Z de l'équation (2.2) donne :

$$E(Z) = S(Z) \cdot (1 + \sum_{k=1}^p \tilde{a}_k Z^{-k}) = S(z) \cdot A(Z) \quad (2.3)$$

$S(Z)$ est la transformée en Z de $s(n)$, $A(Z)$ est l'inverse de $H(Z)$ dans (1.8), d'où il tire son appellation de filtre inverse.

2.1) Méthode d'autocorrélation

Dans cette méthode, l'énergie de l'erreur de prédiction est calculée sur un intervalle infini. Dans la pratique, le signal $s(n)$ est défini sur un intervalle fini, ceci équivaut à multiplier le signal $s(n)$ par une fenêtre de longueur finie correspond à N échantillons.

$$\begin{aligned} s_f(n) &= w(n) \cdot s(n) & 0 \leq n \leq N-1 \\ s_f(n) &= 0 & \text{ailleurs} \end{aligned} \quad (2.4)$$

L'erreur quadratique totale entre le signal prédit et le signal fenêtré est égale à

$$E = \sum_{n=-\infty}^{+\infty} e^2(n) \quad (2.5)$$

$$= \sum_{n=-\infty}^{+\infty} \left(s_f(n) - \sum_{k=1}^p a_k s_f(n-k) \right)^2 \quad (2.6)$$

La minimisation de l'énergie de l'erreur par rapport aux coefficients $a(k)$, ce qui revient à annuler le terme $\frac{\partial E}{\partial a_k}$, conduit au système :

$$\sum_{k=1}^p a_k \sum_{n=-\infty}^{+\infty} s_f(n-i) s_f(n-k) = \sum_{n=-\infty}^{+\infty} s_f(n-i) s_f(n) \quad 1 \leq i \leq p \quad (2.7)$$

L'expression (2.6) peut se mettre sous la forme :

$$\sum_{k=1}^p R(|i-k|) a_k = -R(i) \quad 1 \leq i \leq p \quad (2.8)$$

l'expression (2.7) peut être écrite sous forme matricielle et on aura ainsi :

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_p \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \dots \\ R(p) \end{bmatrix} \quad (2.9)$$

La matrice $p \times p$ obtenu est symétrique et les éléments de chaque diagonale sont identiques, cette matrice est appelée matrice Toeplitz. L'algorithme de Wiener-Livinson-Durbin est utilisé pour résoudre ce système et trouver donc les coefficients a_k

2.2) Méthode de covariance

Dans cette méthode, l'énergie de l'erreur de prédiction est calculée dans un intervalle finie, ceci équivaut à multiplier le signal erreur par une fenêtre de longueur finie de façon que l'énergie à minimiser soit :

$$E = \sum_{n=-\infty}^{+\infty} e(n)^2 \cdot w(n)^2 = \sum_{n=-\infty}^{+\infty} e_f(n)^2 \quad (2.10)$$

La minimisation par rapport aux coefficients a_k conduit, pour $1 \leq k \leq p$, à un système de p équations linéaires :

$$\sum_{k=1}^p \varphi(i,k) a_k = \varphi(i,0) \quad 1 \leq k \leq p \quad (2.11)$$

où $\varphi(i,k)$ est la fonction de covariance et, est définie par :

$$\varphi(i,k) = \sum_{n=-\infty}^{+\infty} w(n)^2 \cdot s(n-i) \cdot s(n-k) \quad (2.12)$$

l'expression (2.11) peut être écrite sous forme matricielle et on aura ainsi :

$$\begin{bmatrix} \varphi(1,1) & \varphi(1,2) & \varphi(1,3) & \dots & \varphi(1,p) \\ \varphi(2,1) & \varphi(2,2) & \varphi(2,3) & \dots & \varphi(2,p) \\ \varphi(3,1) & \varphi(3,2) & \varphi(3,3) & \dots & \varphi(3,p) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \varphi(p,1) & \varphi(p,2) & \varphi(p,3) & \dots & \varphi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_p \end{bmatrix} = \begin{bmatrix} \beta(1) \\ \beta(2) \\ \beta(3) \\ \dots \\ \beta(p) \end{bmatrix} \quad (2.13)$$

où $\beta(i) = \varphi(i,0) \quad 1 \leq i \leq p$

la matrice φ n'est pas une matrice Toeplitz : on a pas en général $\varphi(i,j) = \varphi(|i-j|)$.
La matrice de covariance est symétrique et définie positive, elle peut être décomposée en matrices triangulaires supérieures et inférieures :

$$\varphi = LU \quad (2.14)$$

la décomposition de Cholesky est utilisée pour convertir la matrice de covariance en :

$$\varphi = CC^T \quad (2.15)$$

où $C = L$ et $C^T = U$. les coefficients a_i sont trouvés en résolvant d'abord l'équation :

$$Ly = \psi \quad (2.16)$$

Puis :

$$Ua = y \quad (2.17)$$

2.3) Algorithme de résolution

Nous admettons que la fonction d'autocorrélation $R(k)$ est connue pour $k=0,1,\dots,p$. la moyenne du signal est supposée nulle ; dans le cas contraire, elle est estimée et soustraite.

Il s'agit donc de résoudre le système linéaire (2.9), ce qui revient à inverser une matrice d'ordre « p ». Les méthodes algébriques classiques exigent pour cela un nombre d'opérations (multiplication + addition) de l'ordre de p^3 , ce que l'on note

$O(p^3)$. L'algorithme qui va être décrit profite de la structure particulière de la matrice d'autocorrélation pour résoudre (2.9) par une récursion sur l'ordre de prédiction : autrement dit, ils fournissent toutes les solutions d'ordre $m = 1, 2, \dots, p$, le nombre d'opérations est seulement $O(p^2)$

La variance de l'erreur de prédiction α_p sera obtenue également par une récurrence sur l'ordre m .

Algorithme de Wiener-Levinson-Durbin.

Rappelons que la fonction d'autocorrélation est supposée connue et que pour un signal stationnaire, on a :

$$R(i, j) = R(|i-j|) = R(k) \quad (2.18)$$

Initialement :

$$\alpha_m(0) = 1, \quad (m=1, 2, \dots, p) \quad E_0 = R(0) = \sigma_x^2 \quad (2.19)$$

Récursion:

Pour $m=1, 2, \dots, p$

$$K_m = -\frac{1}{E_{m-1}} \left[R(m) + \sum_{k=1}^{m-1} \alpha_{m-1}(k) R(m-k) \right] \quad (2.20)$$

$$\alpha_m(m) = K_m \quad (2.21)$$

Pour $k=1, 2, \dots, m-1$

$$\alpha_k(m) = \alpha_k(m-1) - K_m \alpha_{m-k}(m-1) \quad (2.22)$$

$$E_m = E_{m-1} (1 - K_m^2) \quad (2.23)$$

Les coefficients $\alpha_k(m)$ résultants, quand $m=p$, représentent les coefficients de prédiction d'un prédicteur linéaire d'ordre p :

$$a_k = \alpha_k(p) \quad (2.24)$$

2.4) La pré-accentuation

L'oreille humaine est plus sensible à la région du spectre du signal parole située au-delà de 1KHz.

La pré-accentuation du signal consiste en un passage dans un filtre de transmittance :

$$H_p(z) = 1 - \mu \cdot Z^{-1} \quad \text{avec } 0.9 < \mu < 1 \quad (2.25)$$

Ceci a pour effet d'accentuer la partie haute fréquence du spectre. Ce pré-traitement se fait avant l'analyse spectrale. Il est utile surtout pour assurer de bonnes conditions pour les algorithmes effectuant l'analyse.

2.5) Considérations pratiques

Pour mener à bien une analyse LPC, il faut pouvoir choisir :

- la fréquence d'échantillonnage f_e ;
- la méthode d'analyse et l'algorithme correspondant;
- l'ordre p de l'analyse LPC;
- le nombre d'échantillons par tranche N et le décalage entre tranches successives L .
- La fonction fenêtre $w(n)$ utilisée.

Le choix de la fréquence d'échantillonnage est fonction de l'application visée et de la qualité du signal à analyser. On choisira plutôt 8 kHz pour les signaux téléphoniques, 10 kHz pour les applications de reconnaissance, et 16 kHz pour les applications de synthèse.

La méthode de covariance ne stipule aucune hypothèse sur la morphologie du signal $s(n)$ en dehors des N points disponibles [4]. Ceci, contrairement à la méthode d'autocorrélation, peut conduire à des solutions instables. Pour cette raison la méthode d'autocorrélation est la plus utilisée. L'algorithme utilisé est celui de Wiener-Livinson-Durbin, car il exige un nombre d'opération (additions + multiplications) de l'ordre de p^2 , contrairement aux algorithmes classiques qui exigent un nombre d'opérations de l'ordre de p^3 .

L'ordre p du système est choisi de façon que l'estimation de l'enveloppe spectrale soit adéquate [5].

La durée des tranches d'analyse et leur décalage sont souvent fixées à 30 et 10 ms respectivement. Ces valeurs ont été choisies empiriquement; elles sont liées au caractère quasi-stationnaire du signal de parole.

Enfin, pour compenser les effets de bord, on multiplie en général préalablement chaque tranche d'analyse par une fenêtre de pondération $w(n)$ de type *fenêtre de Hamming* :

$$\begin{aligned} w(n) &= 0.54 - 0.46 \cdot \cos[2\pi n/(N-1)] && ; 0 \leq n \leq N-1 \\ w(n) &= 0 && ; \text{ailleurs} \end{aligned} \quad (2.26)$$

La figure (2.1) donne une représentation complète de la l'analyse LPC sous forme d'organigramme.

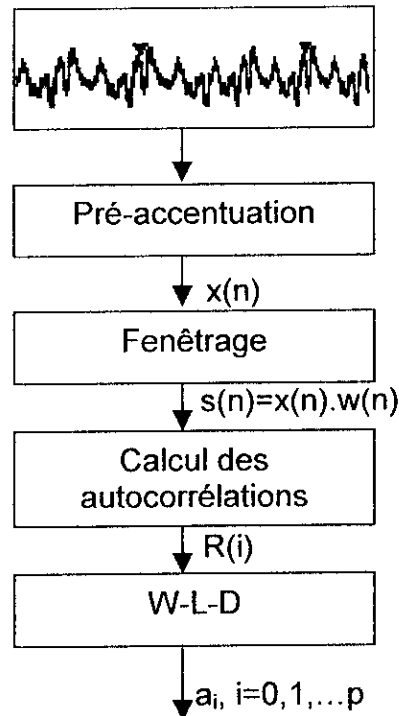


Figure (2.1) Organigramme de l'analyse LPC

3) Analyse du signal vocal

Cette étape consiste à extraire les paramètres du signal parole qui sont :

- les coefficients a_i ,
- le signal erreur (résidu).

3.1) Les coefficients a_i

Les coefficients a_i sont obtenus en utilisant l'analyse LPC, mais leur utilisation directe n'est pas recommandée, car la quantification de ces coefficients peut induire à une instabilité du filtre de synthèse. Plusieurs solutions ont été proposées, telles que les coefficients de réflexion et les paramètres LSF_i (Line Spectral Frequencies), pour palier à ce problème. Le modèle du codeur LPC utilise la représentation en LSF.

Calcul des LSF

Le filtre inverse $A(z)$ associé au filtre de synthèse $H(z)$ satisfait la relation de récurrence suivante [4]:

$$A_n(z) = A_{n-1}(z) - K_n z^{-n} A_{n-1}(z^{-1}) ; n=1, 2, \dots, p \quad (2.27)$$

Avec $A_0(z) = 1$, et K_n est le $n^{\text{ième}}$ coefficient de réflexion. En étendant l'ordre du filtre à $n=p+1$, l'équation (2.27) devient :

$$A_{p+1}(z) = A_p(z) - K_{p+1}z^{-(p+1)}A_p(z^{-1}) \quad (2.28)$$

On considère deux conditions limites artificielles $K_{p+1}=1$ et $K_{p+1}=-1$, qui correspondent, respectivement, à l'ouverture complète au niveau de la glotte dans le modèle du tube acoustique [4]. On obtient donc deux polynômes :

$$F_1(z) = A_p(z) + z^{-(p+1)}A_p(z^{-1}) \quad (2.29)$$

$$F_1(z) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_2z^{-(p-1)} + a_1z^{-p} + z^{-p} \quad (2.30)$$

qui correspond à $K_{p+1}=-1$, et

$$F_2(z) = A_p(z) - z^{-(p+1)}A_p(z^{-1}) \quad (2.31)$$

$$F_2(z) = 1 + a_1z^{-1} + a_2z^{-2} + \dots - a_2z^{-(p-1)} - a_1z^{-p} - z^{-p} \quad (2.32)$$

qui correspond à $K_{p+1}=1$

Ces deux polynômes possédant les propriétés suivantes :

- Toutes les racines de $F_1(z)$ et $F_2(z)$ se situent sur le cercle unité, et elles sont données par $e^{j2\pi f_i}$.
- Les racines de $F_1(z)$ et $F_2(z)$ alternent sur le cercle unité.
- Le filtre inverse $A(z)$ est à phase minimale.

Dans le cas d'un ordre de prédiction p pair, les deux polynômes $F_1(z)$ et $F_2(z)$ sont données par :

$$F_1(z) = (1+z^{-1}) \prod_{i=2,4,\dots,p} (1-2\cos(2\pi f_i)z^{-1}+z^{-2}) \quad (2.33)$$

$$F_2(z) = (1-z^{-1}) \prod_{i=1,3,\dots,p-1} (1-2\cos(2\pi f_i)z^{-1}+z^{-2}) \quad (2.34)$$

Les paramètres f_i sont appelés les paires de raies spectrales LSP(Line Spectrum Pairs) ou LSF (Line Spectral Frequencies).

Les paramètres LSF peuvent être interprétés comme les fréquences de résonances du conduit vocal

3.2) Le signal erreur (résidu)

Le signal erreur (résidu) peut être extrait à partir du signal vocal, il suffit d'appliquer ce dénier à l'entrée d'un filtre inverse

4) Reconstitution du signal vocal

Cette étape se fait à la réception, elle consiste à synthétiser le signal vocal à partir de ses paramètres qui sont :

- les coefficients a_i qui sont calculés à partir des paramètres LSF_i
- l'erreur de prédiction.

4.1) Calcul des coefficients a_i

La conversion des paramètres LSF_i en coefficients de prédiction est moins complexe que le calcul des LSF_i à partir des coefficients a_i [4]. En utilisant la relation (2.22) :

$$A(z) = (F_1(z) + F_2(z))/2 \quad (2.35)$$

Les polynômes $F_1(z)$ et $F_2(z)$ sont obtenus à partir des coefficients LSF_i en utilisant les équations (3.33) et (3.34).

5) Codage et Décodage des paramètres LSF_i

La quantification vectorielle est une opération qui permet de représenter une valeur x à k composants par un vecteur y appartenant à un ensemble fini de M vecteurs (dictionnaire)[3]. Si on remplace l'ensemble des paramètres du filtre à court-terme par un des membres d'un dictionnaire, alors seul l'index de ce vecteur est à transmettre pour gagner sur le débit de transmission.

6) Modèle du codeur LPC

La figure (2.2) donne le schéma de principe d'un codeur LPC. Le signal vocal mesuré par un microphone est échantillonné, analysé par la prédiction linéaire pour obtenir les coefficients a_i . Ces coefficients sont transformés en coefficients LSF_i qui sont ensuite quantifiés, c'est à dire qu'ils sont codés sur un ensemble fini de nombres entiers (ce qui permet d'associer à chaque paramètre un nombre fini de bits par trame) puis transmis avec le résidu. Le récepteur reçoit les coefficients LSF_i codés, il les décode, les transforme en coefficients a_i puis produit un signal de synthèse, image du signal vocal.

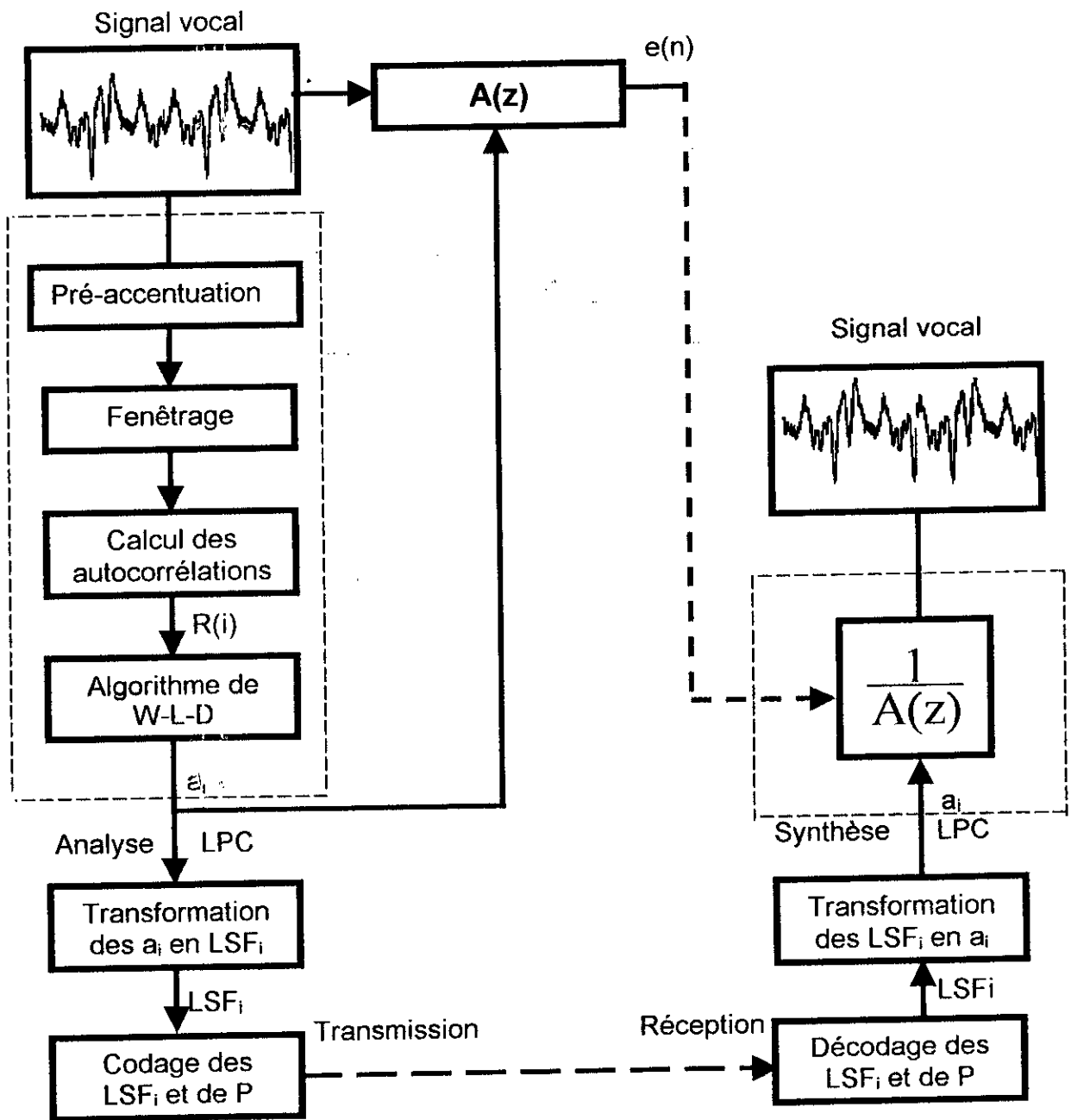


Figure (2.2) schéma synoptique du codeur LPC.

7) Conclusion

Dans le codeur LPC, l'opération de codage consiste en trois étapes :

- Analyse du signal vocal qui consiste à décomposer le signal à coder en un ensemble de paramètres.
- Quantification des coefficients LSF_i en utilisant des dictionnaires (seule l'indice du dictionnaire est à transmettre).
- Synthèse du signal vocal à partir des paramètres du modèle.

Les performances du codeur dépendent essentiellement du modèle et du dictionnaire.

III) ETUDE DU DSP TMS320C31.....	5
1) Introduction.....	24
2) Classes de signaux à traiter.....	24
3) Domaines d'application des DSP.....	24
3.1) Domaine des télécommunications.....	25
3.2) Domaine militaire.....	25
3.3) Domaine des multimédia et du grand public.....	25
3.4) Domaine médical.....	25
3.5) Domaine de l'automatique.....	25
3.6) Domaine de l'instrumentation.....	25
4) Architecture.....	25
4.1) Description générale.....	25
4.2)Présentation matérielle.....	25
4.3) Description et rôle des signaux.....	26
4.4) Diagramme bloc du C31.....	28
4.4.1) Le Multiplieur.....	28
4.4.2) L'unité arithmétique et logique (UAL)	28
4.4.3) L'unité de calcul des registres Auxiliaires (ARAU)	29
4.4.5) Les Registres.....	29
• Les registres de précision étendue (R0-R7).....	29
• Les registres auxiliaires (AR0-AR7)	29
• Les registres d'index (IR0-IR1)	29
• Le pointeur de pile système (SP)	29
• Le registre d'état (ST)	29
• Le registre d'interruption CPU/DMA (IE)	30
• Le registre indicateur d'interruption CPU (IF)	31
• Le registre indicateur d'E/S (IOF)	31
• Les registres de répétition de bloc (RC-RS-RE)	32
• Le compteur programme (PC)	32
• La mémoire interne.....	32
4.5) Les périphériques.....	32
4.5.1) Le timer.....	32
• Le registre de contrôle globale (GCR)	33
• Le registre de période.....	34
• Le registre du comptage.....	34
4.5.2) Le port série.....	34
• Le registre de contrôle globale (GCR)	35
• Le registre de contrôle des broche FSX/DX/CLKX.....	37
• Le registre de contrôle des broche FSR/DR/CLKR.....	37
• Le registre de contrôle du timer de réception/transmission.....	39
• Le registre de comptage du timer transmission/réception.....	40

- Le registre période du timer transmission/réception.....40
- Le registre de transmission de données.....40
- Le registre de réception de données.....41
- La carte mémoire du DSP TMS320C31.....42
- La carte mémoire des périphériques.....43

5) Les modes d'adressage.....43

5.1) Adressage par registres.....44

5.2) Adressage direct.....44

5.3) Adressage immédiat.....44

5.4) Adressage relatif PC.....44

5.5) Adressage indirect.....44

5.5.1) Adressage indirect avec déplacement.....44

5.5.2) Adressage indirect avec registres d'index.....45

5.5.3) Adressage indirect (cas spéciaux).....45

5.6) Adressage circulaire.....45

5.6) Adressage par inversion de bits.....46

6) Conclusion.....47

1) Introduction

Les processeurs de traitement du signal ont vu leur utilisation se développer considérablement depuis 1985 et, ce, notamment avec l'évolution des télécommunications ; actuellement de nombreux produits comportent un Digital Signal Processor (DSP) [8].

2) Classes de signaux à traiter

Le nombre de temps de cycles n du DSP est définie comme étant le rapport entre la fréquence d'horloge du microprocesseur F_C et la fréquence d'échantillonnage F_E .

La fréquence d'échantillonnage F_e est fonction du domaine d'applications et du type de signal à traiter, elle est choisie de façon à respecter le théorème de Shannon :

$$F_E \geq 2.F_{\max} \quad (3.1)$$

Le tableau suivant résume les cas les plus courants :

Signal à traiter	Fréquence d'échantillonnage	Période d'échantillonnage	Nombre de temps de cycle T_c entre deux échantillons avec $T_c = 50$ ns
Signaux vibratoires	100 kHz	10 μ s	200
Parole	8 kHz	125 μ s	2 500
Audio	48 kHz	20,8 μ s	417
Vidéo	10 MHz	100 μ s	2

Tableau (3.1) classe des signaux à traiter [8]

A l'examen de ce tableau. On comprend que les DSP sont couramment utilisés pour traiter en temps réel les signaux de la bande audio (0 à 20 kHz de largeur de bande), mais que le traitement des signaux vidéo en temps réel reste la plupart du temps du domaine des circuits intégrés dédiés ou d'architecture multi-DSP [8].

3) Domaines d'application des DSP

A l'origine, les deux principaux domaines d'applications des DSP ont été les télécommunications et le secteur militaire. Aujourd'hui, les applications se sont diversifiées et s'orientent vers le domaine médical, le multimédia et du grand public, l'automatique, l'instrumentation ... etc.

3.1) Domaine des télécommunications

- La téléphonie filaire : les modems, les codeurs de parole de type ADPCM, les multiplexeurs, les annuleurs d'écho, les télécopieurs.
- La radiotéléphonie ; téléphone sans fil (les codeurs de parole GSM, les modems radio).

3.2) Domaine militaire

- Le guidage de missiles, la navigation, les modems radio, les communications cryptées, le traitement des signaux radars et sonars.

3.3) Domaine des multimédia et du grand public

- La compression des images fixes ou animées, la radiodiffusion sonore numérique, les cartes multimédia pour PC ou stations de travail, la synthèse musicale.

3.4) Domaine médical

- La compression d'images médicales (radiologie, échographie) en vue de leur archivage, le traitement des signaux biophysiques ECG, EEG, les implants cochléaires.

3.6) Domaine de l'automatique

- La surveillance et la commande de machines, le contrôle des moteurs, les robots.

3.7) Domaine de l'instrumentation

- Les analyseurs de spectres, les générateurs de fonctions, l'interprétation des signaux sismiques.

4) Architecture

4.1) Description Générale

Le processeur TMS320C31 fait partie de la troisième génération des processeurs de Texas Instrument de la famille TMS320C3X. Du point de vue rapidité. Ses caractéristiques sont les suivantes :

- Temps d'exécution d'une instruction : 60 ns.
- 33,3 MFLOPS (million floating-point opérations per second).
- 16,7 MIPS (million instructions per second).

4.2) Présentation Matérielle

Le TMS320C31 est présenté dans un boîtier de 132 broches.

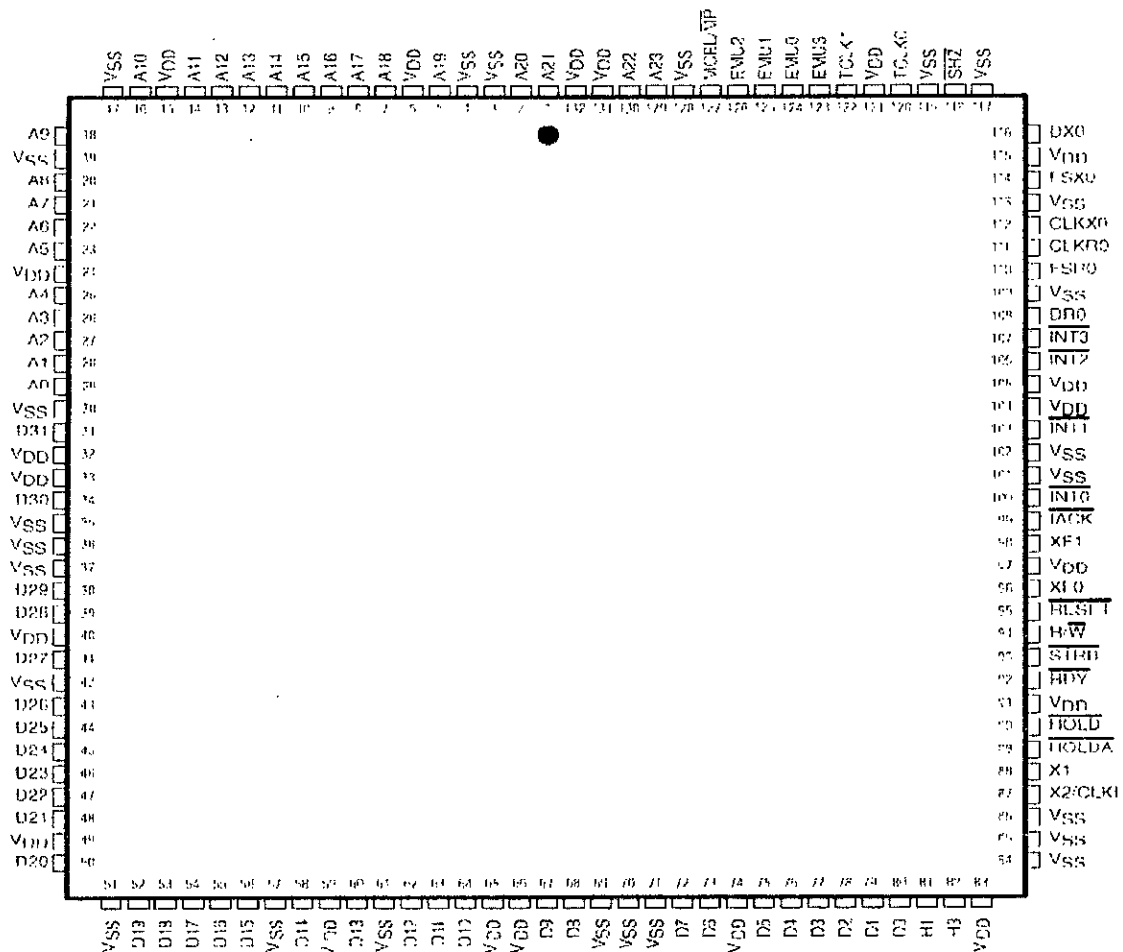


Figure (3.1) boîtier du TMS320C31 [17]

4.3) Description Et Rôle Des Signaux :

Terminal Nom	QTY	Type	Description
Bus d'interface primaire			
D ₀ -D ₃₁	32	E/S/Z	Port de donnée 32-Bit
A ₀ -A ₂₃	24	S/Z	Port d'adresse 24-Bit
R/W	1	S/Z	Read/write. Indique le sens de l'échange entre le circuit et un circuit externe.
STRAB	1	S/Z	External-access strobe
RDY	1	E	Entrée indique la validité de la donnée tant qu'il est bas, le dispositif externe n'est pas prêt et le DSP attend un cycle supplémentaire et test le RDY
HOLD	1	E	Signal d'entrée indiquant une demande de contrôle par un dispositif externe des bus données, adresse et contrôle.
HOLDA	1	S/Z	Ce signal est envoyé pour indiquer au circuit externe que le DSP est en mode <i>hold</i> et que les bus d'adresse et de donnée et les lignes de contrôle correspondantes sont en haute impédance.

Signaux de contrôle			
RESET	1	E	Reset. Lorsque RESET est à l'état logique haut, les dispositifs sont réinitialisés.
INT0-INT3	4	E	Interruptions externes
IACK	1	S/Z	Interrupt acknowledge. L'interruption est reçue et le processeur accède au vecteur d'interruption.
MCBL/MP	1	E	Microcomputer boot-loader/microprocessor mode-select
SHZ	1	E	Shutdown high impedance. Fermeture de la haute Impédance
XF0-XF1	2	E/S/Z	External flags. XF1 et XF2 sont utilisés en tant qu' E/S comme raison générale
Signaux du port série			
CLKR0	1	E/S/Z	Serial port 0 receive clock. (horloge de réception de données vers les registres de réception du port série)
CLKT0	1	E/S/Z	Serial port 0 transmit clock. (horloge de transmission série)
DR0	1	E/S/Z	Data-receive. (Port de réception série)
DX0	1	E/S/Z	Data-transmit output. (port de transmission série)
FSR0	1	E/S/Z	Frame-synchronization pulse for receive. (synchronisation de trame pour la réception)
FSX0	1	E/S/Z	Frame-synchronization pulse for transmit. (synchronisation de trame pour la transmission)
Signaux d'horloge			
TCLK0	1	E/S/Z	Elle est utilisée, par le timer 0, comme entrée/sortie d'horloge ou d'impulsion, ou comme une broche d'E/S à usage générale
TCLK1	1	E/S/Z	Elle est utilisée, par le timer 1, comme entrée/sortie d'horloge ou d'impulsion, ou comme une broche d'E/S générale.
Signaux d'alimentation et d'oscillation			
H1	1	S/Z	Horloge externe. La période de H1 égale deux fois TCLKIN (horloge interne).
H3	1	S/Z	Horloge externe. Le période de H3 égale deux fois TCLKIN (horloge interne).
VDD	20	E	Pour C31 l'alimentation est 5 V, et pour LC31 est 3,3 V. doit être connecté au courant d'alimentation. ¹
VSS	25	E	Cette broche doit être connectée au niveaux de la terre (la masse) d'alimentation.
X1	1	S	La sortie vers l'oscillateur quartz. Si le quartz ne marche pas X1 est déconnecté.
X2/CLKIN	1	E	Entrée de l'oscillateur interne (quartz) .
Signaux de l'émulation ²			
EMU0-EMU23		E	Réservé pour l'émulation
EMU3	1	S/Z	Réservé pour l'émulation

Tableau (3.2) description et rôle des signaux.

1 : il est conseillé de coupler avec un condensateur de 0,1µF.

2 : pour réserver les pins, on utilise des résistances de 18 kΩ à 22 kΩ.

4.4) Diagramme bloc de C31

functional block diagram

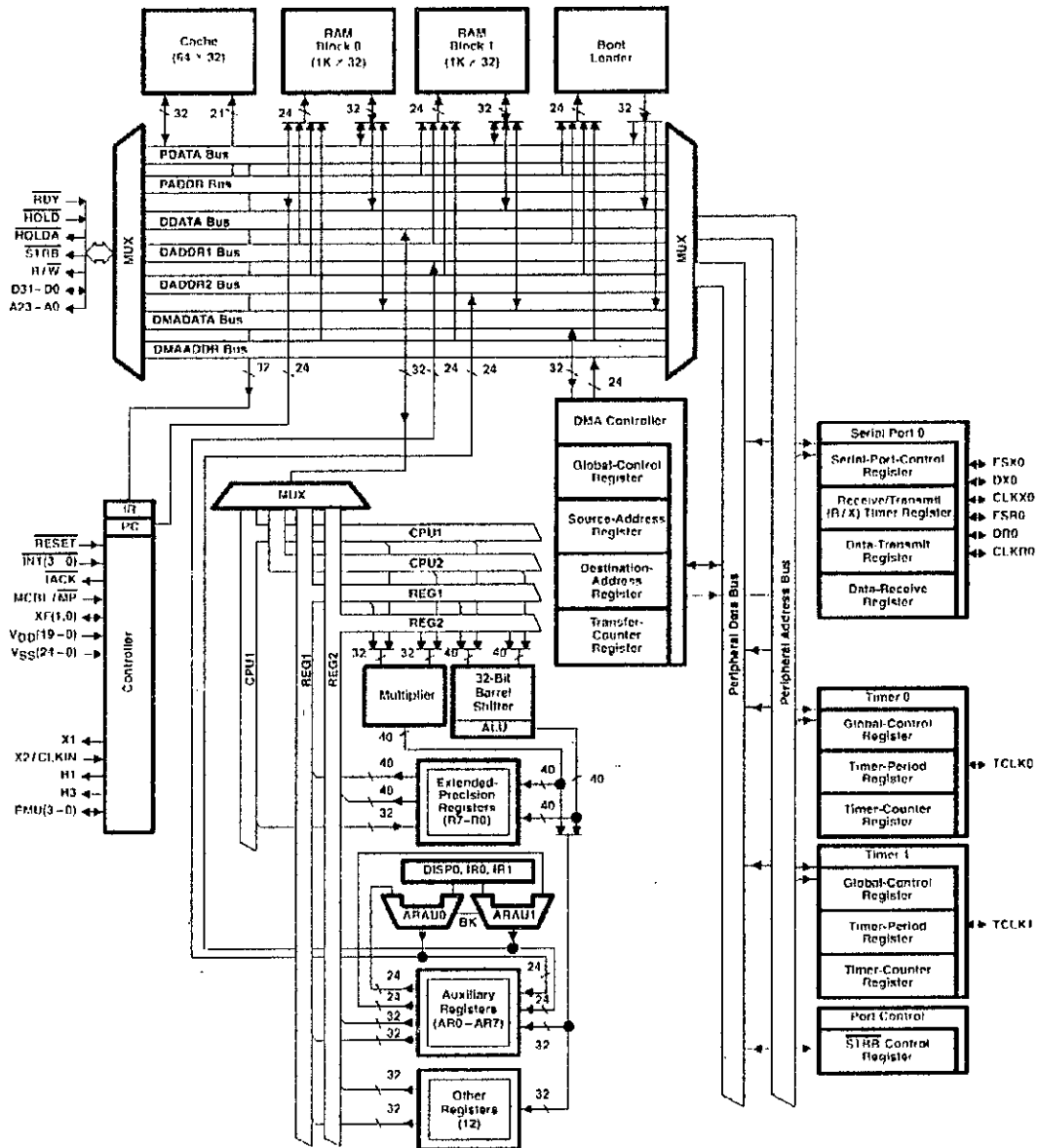


Figure (3.2) Diagramme bloc du C31[10]

4.4.1) Le Multiplieur

Le multiplieur effectue la multiplication signée de deux nombres en virgule fixe ou flottante en un seul temps de cycle. En virgule fixe les deux opérands sont sur 24 bits et le résultat est sur 32 bits. En virgule flottante les deux opérands sont sur 32 bits et le résultat est sur 40 bits.

4.4.2) L'unité arithmétique et logique (UAL)

L'UAL travaille sur deux opérands 32 bits (en virgule fixe) ou 40 bits (en virgule flottante) ainsi que sur des valeurs logiques 32 bits. Le calcul se fait en complément à deux.

4.4.3) L'unité de calcul des registres Auxiliaires (ARAU)

Cette unité effectue des opérations arithmétiques sur les registres auxiliaires. Le DSP TMS320C31 possède deux ARAUs qui travaillent en parallèle avec l'UAL et le multiplieur.

4.4.4) Les Registres

Le TMS320C31 possède 29 registres de 32 bits sauf pour les registres Rn.

- **Les registres de précision étendue (R0-R7)**

Ces registres 40 bits sont dédiés aux calculs arithmétiques ainsi qu'au sauvegarde intermédiaire des résultats de calculs. En virgule fixe, seulement les bits 0-31 sont utilisés, alors qu'en virgule flottante, l'ensemble des bits 0-39 sont employés.

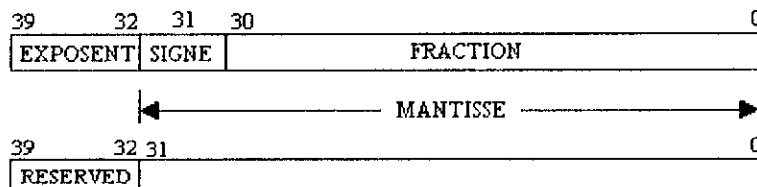


Figure (3.3) représentation en virgule fixe et flottante [10]

- **Les registres auxiliaires (AR0-AR7)**

Ces registres de 32 bits sont principalement utilisés dans l'adressage mémoire, et peuvent être utilisés comme des registres à usage général.

- **Les registres d'index (IR0-IR1)**

Ces registres 32 bits sont utilisés par l'unité de calcul des registres auxiliaires pour calculer une adresse indexée.

- **Le pointeur de pile système (SP)**

Ce registre 32 bits pointe sur le dernier élément mis en pile. Il est utilisé par le processeur lors des appels de sous-programmes des routines d'interruptions.

- **Le registre d'état (ST)**

C'est un registre 32 bits, il contient l'ensemble des informations relatives à l'état du CPU. Il est modifié par les opérations arithmétiques ou logiques.

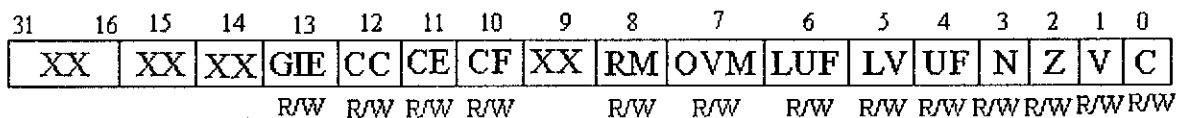


Figure (3.4) le registre d'état [10]

Bit Name	Name	Description
C	Carry flag	Ce bit passe à 1 si une retenue est générée dans une addition ou une soustraction.
V	Overflow flag	Ce bit indique un dépassement lors d'un calcul.
Z	Zéro flag	Z est mis à 1 lorsque le résultat d'une opération est nul
N	negative flag	N indique que le résultat est négatif.
UF	Floting-point underflow flag	floting-point underflow condition flag
LV	Latched overflow flag	latched overflow condition flag
LUF	Latched floating-point Underflow flag	latched floating-point underflow condition flag
OVM	Over flow flag <i>mode de flag</i>	Ce bit définit si lors d'un dépassement le résultat est saturé ou non.
RM	Repeat mode flag	Repeat mode flag. Ce bit est mis à 1 lorsque le DSP exécute une opération de répétition
CE	Cache enable	Ce bit active ou désactive la mémoire cache.
CF	Cache freeze	Ce bit active ou désactive la mémoire cache. CE CF Effet 0 0 Cache not enabled 0 1 Cache not enabled 1 0 Cache enabled and not frozen 1 1 Cache enabled but frozen (cache read only)
CC	Cache clear	CC = 1 invalidates all entries in the cache. This bit is always cleared after it is written to, and is always read as 0. At reset, 0 is written to this bit.
GIE	Global interrupt enable	Si GIE = 1, les interruptions du DSP sont activées Si GIE = 0, les interruptions du DSP sont désactivées

Tableau (3.3) description de chaque bit du registre d'état

• Le registre d'interruption CPU/DMA (IE)

C'est un registre 32 bits qui permet le contrôle de toutes les interruptions CPU qui correspondent aux bits 0 à 10, et les interruptions DMA qui correspondent aux bits 16 à 26.

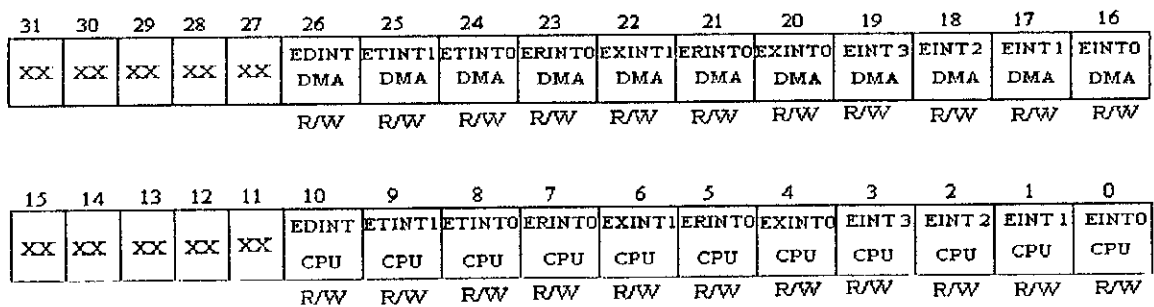


Figure (3.5) le registre IE [10]

Abbreviation	Description
EINT0 (CPU)	CPU external interrupt 0 enable
EINT1 (CPU)	CPU external interrupt 1 enable
EINT2 (CPU)	CPU external interrupt 2 enable
EINT3 (CPU)	CPU external interrupt 3 enable
EXINT0 (CPU)	CPU serial port 0 transmit interrupt enable
ERINT0 (CPU)	CPU serial port 0 receive interrupt enable
ETINT0 (CPU)	CPU timer0 interrupt enable
ETINT1 (CPU)	CPU timer1 interrupt enable
EDINT (CPU)	CPU DMA controller interrupt enable
EINT0 (DMA)	DMA external interrupt 0 enable
EINT1 (DMA)	DMA external interrupt 1 enable
EINT2 (DMA)	DMA external interrupt 2 enable
EINT3 (DMA)	DMA external interrupt 3 enable
EXINT0 (DMA)	DMA serial port 0 transmit interrupt enable
ERINT0 (DMA)	DMA serial port 0 receive interrupt enable
ETINT0 (DMA)	DMA timer0 interrupt enable
ETINT1 (DMA)	DMA timer1 interrupt enable
EDINT (DMA)	DMA controller interrupt enable

Tableau (3.4) description de chaque bit du registre IE

- **Le registre indicateur d'interruption CPU (IF)**

C'est un registre 32 bits indiquant la présence d'interruption

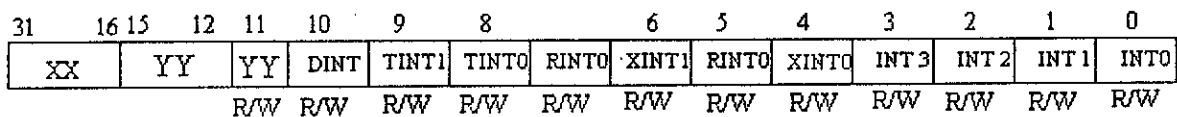


Figure (3.6) Le registre indicateur d'interruption CPU [10]

Bit Name	Reset Value	Function
INT0	0	External interrupt 0 flag
INT1	0	External interrupt 1 flag
INT2	0	External interrupt 2 flag
INT3	0	External interrupt 3 flag
XINT0	0	Serial port 0 transmit flag
RINT0	0	Serial port 0 receive flag
TINT0	0	Timer 0 interrupt flag
TINT1	0	Timer 1 interrupt flag
DINT	0	DMA channel interrupt flag

Tableau (3.5) Signification de chaque bit du registre IF

- **Le registre indicateur d'E/S (IOF)**

Ce registre 32 bits permet de contrôler la fonction des deux broches XF0 et XF1.

- **Les registres de répétition de bloc (RC-RS-RE)**

Ces trois registres sont utilisés lorsque le DSP travaille dans le "repeat mode"

- a) **Le registre RC** : ce registre 32 bits contient le nombre de fois qu'un bloc de code doit être répété.
- b) **Le registre RS** : ce registre 32 bits contient l'adresse de début du bloc à répéter
- c) **Le registre RE** : ce registre 32 bits contient l'adresse de fin du bloc à répéter.

- **Le compteur programme (PC)**

Ce registre 32 bits contient l'adresse de la prochaine instruction à exécuter.

4.4.5) La mémoire interne

Elle est formée de plusieurs blocs, et comprend :

- deux blocs de mémoire RAM de 1 kmots (de 32 bits) ;
- un bloc de mémoire ROM de 4 kmots (de 32 bits) ;
- un cache instructions de 64 mots (de 32 bits) utilisée pour stocker les parties de code les plus utilisées.

4.5) Les périphériques

Le DSP TMS320C3X disposent de plusieurs périphériques. Le C31 comprend deux timers, un port série, Un contrôleur DMA

4.5.1) Le timer

Le timer , permet comme la plupart des temporisateurs, aussi bien de produire des signaux impulsionnels que de compter des événements. Le signal d'horloge de fréquence programmable peut être utilisé comme signal d'horloge de référence du DSP.

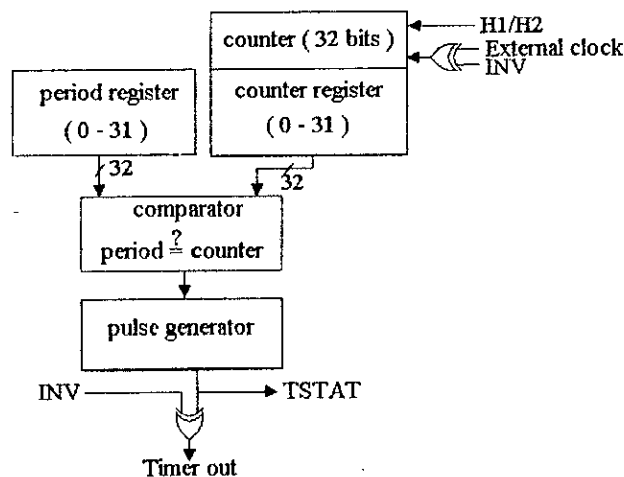


Figure (3.7) diagramme bloc du timer [10]

Le timer contient trois registres qui permettent de le contrôler :

- **Le registre de contrôle global (GCR)**

Ce registre 32 bits permet de commander le mode de fonctionnement du timer. Pour accéder au contenu de ce registre, dont la structure est présentée sur la figure (3.8), on utilisera les instructions de chargement.



Figure (3.8) Le registre de contrôle global du timer [10]

Abbreviation	Nom	Description
FUNC	Function	Ce bit contrôle la fonction de TCLK. si func=0 TCLK est configuré comme un port d'entrée/sortie, dans le cas contraire (func=1) , c'est le bit CLKSRC qui détermine d'horloge du timer.
I/O	Input/output	Si FUNC = 0 et CLKSRC = 0, TCLK est configuré comme un port d'E/S. Si I/O = 0, TCLK est configuré comme un port d'entrée. Si I/O = 1, TCLK est configuré comme un port de sortie
DATOUT	Data output	Contrôle TCLK lorsque 'C3x est configuré comme un port d'E/S.
DATIN	Data input	Entrée de données sur TCLK ou DATOUT.
GO	Go	Réinitialise le comptage. lorsque GO = 1 et le timer est activé, le compteur est mis à zéro et commence de s'incrémenter. GO=0 n'a pas d'effet.
HLD	Counter hold signal	Lorsque ce bit est à zéro le compteur est désactivé. Dans le cas contraire le timer est réinitialisé (il faut que GO =1 aussi).
C/P	Clock/pulse mode control	lorsque ce bit est à 1, le mode horloge est choisi, dans le cas contraire, le mode sortie impulsion est choisi
CLKSRC	Clock source	Ce bit permet de déterminer la source d'horloge qui peut être interne ou externe. Si CLKSRC =1 (source interne), un signal d'horloge de fréquence égale à la moitié de la fréquence du DSP et est utilisé pour incrémenter le compteur. Dans le cas contraire (source externe, CLKSRC =0) la fréquence d'horloge du timer est inférieure à la moitié de la fréquence du DSP et dans ce cas le timer délivre un signal impulsionnel.

INV	Inverter control bit	Lorsque ce bit est mis à 1, le signal d'horloge est inversé
TSTAT	Timer status bit	Ce bit indique l'état du timer.

Tableau (3.6) description des bits du registre (GCR)

- **Le registre de période**

Il permet de spécifier (en mode timer) la fréquence du signal généré par le timer.

- **Le registre du comptage**

En mode timer, le contenu de ce registre est incrémenté à chaque top d'horloge puis comparé au contenu du registre de période. Lorsque les deux contenus sont égaux, une impulsion est générée via la broche TCLK.

4.5.2) Le port série

Le port série est un périphérique qui permet la communication avec des circuits externes tels que des convertisseurs A/N ou avec d'autres processeurs. Le port série peut transmettre ou recevoir des mots de 8,16,24 ou 32 bits en mode synchrone (mode standard) ou asynchrone (mode continu). L'horloge du port série peut être interne grâce à son timer ou externe via les broches CLKX en transmission et CLKR en réception [16]. Le port série dispose de huit registres qui sont :

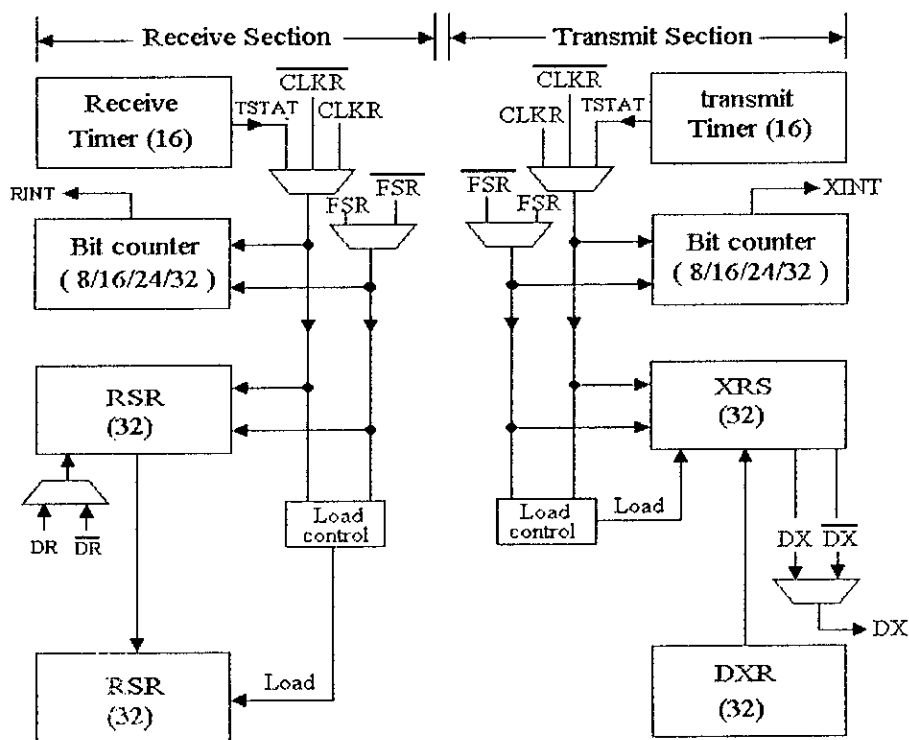


Figure (3.9) diagramme bloc du port série [10]

• Le registre de contrôle global (GCR)

Ce registre fixe les modes de fonctionnement du port série.

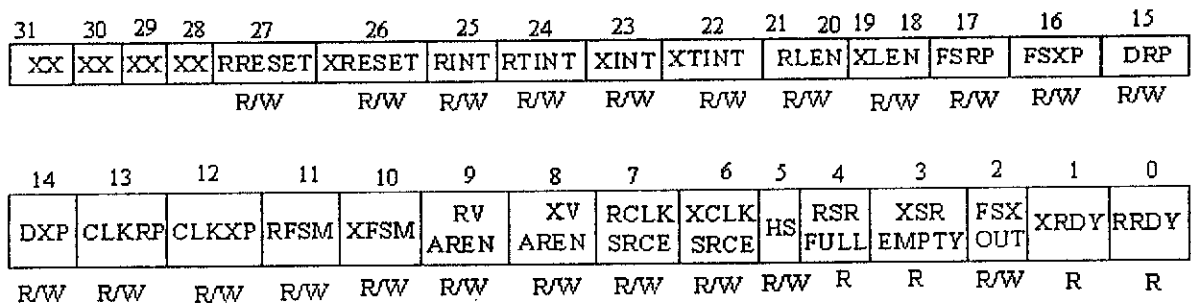


Figure (3.10) Le registre GCR du port série [10]

Abbreviation	Nom	Description
RRDY	Receive ready flag	Une transition de 0 à 1 du RRDY indique qu'il est possible de lire les données, une interruption de réception est générée lors de cette transition.
XRDY	Transmit ready flag	Une transition de 0 à 1 du XRDY indique qu'il est possible de transmettre les données, une interruption de transmission est générée lors de cette transition.
FSXOUT	Transmit frame sync configuration	Si FSXOUT = 0 la broche FSX est configurée comme une entrée. FSXOUT = 1 configure la broche FSX comme une sortie
XSREEMPTY	Transmit-shift register empty flag	Si XSREEMPTY = 0, le registre à décalage de transmission est vide. Si XSREEMPTY = 1, il n'est pas encore vide.
RSRFULL	Receive shift register full flag	RSRFULL = 1, indique que le récepteur a subi un <i>overrun</i> (écrasement). Dans le mode continu RSRFULL est mis à 1 lorsque RSR et DRR sont pleins. Dans le mode discontinu, RSRFULL est mis à 1 lorsque RSR et DRR sont pleins et un nouveau FSR est reçu.
HS	Handshake	Si HS = 1, le mode handshake est activé. Si HS = 0, le mode handshake est désactivé
XCLK SRCE	Transmit clock source	Si XCLK SRCE = 1, l'horloge interne en émission est utilisée. Si XCLK SRCE = 0, l'horloge externe en émission est utilisée.
RCLK SRCE	Receive clock source	Si RCLK SRCE = 1, l'horloge interne en réception est utilisée. Si RCLK SRCE = 0, l'horloge externe en réception est utilisée.

XVAREN	Transmit data rate mode	Ce bit indique si le débit de transmission est fixe ou variable. Pour un débit fixe, FSX est activé pour au moins un cycle de XCLK et devient inactive avant que la transmission commence. Pour un débit variable, FSX est activée tant que tout les bits sont en transmission.
RVAREN	Receive data rate mode	Ce bit indique si le débit de réception est fixe ou variable. Pour un débit fixe, FSX est activé pour au moins un cycle de XCLK et devient inactive avant que la réception commence. Pour un débit variable, FSX est activée tant que tout les bits sont en réception.
XFSM	Transmit frame sync mode	Configure le port pour le mode continu. Si XFSM = 1 (mode continu), seule le premier mot est transmis avec une impulsion de synchronisation, et le reste sont transmis continuellement jusqu'à la fin de la transmission. Si XFSM = 0 (mode standard), chaque mot est associé à une impulsion de synchronisation.
RFSM	Receive frame sync mode	Configure le port pour le mode continu Si RFSM = 1 (mode continu), seule le premier mot est reçu avec une impulsion de synchronisation, et le reste sont transmis continuellement jusqu'à la fin de la réception. Si RFSM = 0 (mode standard), chaque mot est associé à une impulsion de synchronisation
CLKXP	CLKX polarity	Si CLKXP = 0, CLKX est activé à l'état haut. Si CLKXP = 1, CLKX est activé à l'état bas.
CLKRP	CLKR polarity	Si CLKRP = 0, est activé à l'état haut. Si CLKRP = 1, est activé à l'état bas.
DXP	DX polarity	Si DXP = 0, DX est activé à l'état haut. Si DXP = 1, est activé à l'état bas
DRP	DR polarity	Si DRP = 0, est activé à l'état haut. Si DRP = 1, est activé à l'état bas
FSXP	FSX polarity	Si FSXP = 0, est activé à l'état haut. Si FSXP = 1, est activé à l'état bas
FSRP	FSR polarity	Si FSRP = 0, est activé à l'état haut. Si FSRP = 1, est activé à l'état bas
XLEN	Transmit word length	Ces deux bits indiquent la longueur du mot en transmission : 0 0 — 8 bits. 0 1 — 16 bits. 1 0 — 24 bits. 1 1 — 32 bits.
RLEN	Receive word length	Ces deux bits indiquent la longueur du mot en réception : 0 0 — 8 bits. 0 1 — 16 bits. 1 0 — 24 bits. 1 1 — 32 bits.

XTINT	Transmit timer interrupt enable	Si XTINT = 0, l'interruption du timer en transmission est désactivée. Si XTINT = 1, l'interruption du timer en transmission est activée.
XINT	Transmit interrupt enable	Si XINT = 0, l'interruption en transmission est désactivée. Si XINT = 1, l'interruption en transmission est activée.
RTINT	Receive timer interrupt enable	Si RTINT = 0, l'interruption du timer en réception est désactivée. Si RTINT = 1, l'interruption du timer en réception est activée.
RINT	Receive interrupt enable	Si RINT = 0, l'interruption en réception est activée. Si RINT = 1, l'interruption en réception est activée.
XRESET	Transmit reset	Ce bit effectue le RESET de l'émetteur
RRESET	Receive reset	Ce bit effectue le RESET du récepteur.

Tableau (3.7) description des bits du registre GCR

• le registre de contrôle des broches FSX/DX/CLKX

Ce registre contrôle la fonction des broches FSX, DX et CLKX.

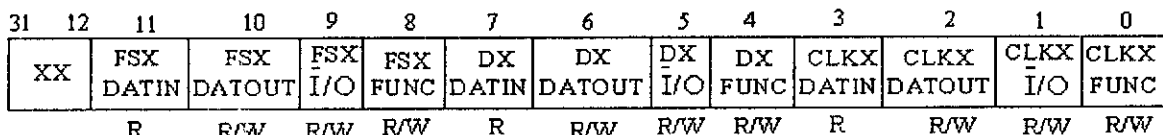


Figure (3.11) le registre de contrôle des broches FSX,DX,CLKX [10]

• Le registres de contrôle des broches FSR/DR/CLKR

Ce registre contrôle la fonction des broches FSR, DR et CLKR.

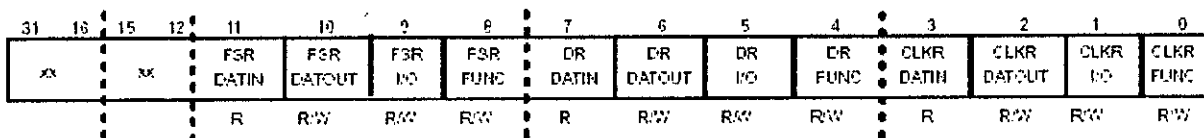


Figure (3.12) Le registre de contrôle des broches FSR/DR/CLKR [10]

Abréviation	Nom	Description
CLKX (CLKR)FUNC	Clock transmit (receive) Function	Ce bit contrôle la fonction CLKX (CLKR).
CLKX (CLKR) I/O	Clock transmit (receive) input/output mode	Si CLKX (CLKR) I/O = 0, CLKX(CLKR) est configuré comme un port d'entrée If CLKX(CLKR) I/O = 1, CLKX(CLKR) est configuré comme un port de sortie
CLKX(CLKR) DATOUT	Clock transmit (receive)	CLKX(CLKR) est la sortie de données lorsqu'il est configuré comme sortie.
CLKX(CLKR) DATIN	Clock transmit (receive) data input	CLKX(CLKR) est la l'entrée de données lorsqu'il est configuré comme entrée.
DX (DR) FUNC	DX (DR) function	DX (DR) FUNC contrôle la fonction de DX (DR). Si DX (DR) FUNC = 0, DX (DR) est configuré comme un port d'entrée/sortie. If DX (DR) FUNC = 1, DX (DR) est configuré comme un port série.
DX (DR) I/O	DX (DR) input/output mode	Si DX (DR) I/O = 0, DX (DR) est configuré comme un port d'entrée. Si DX (DR) I/O = 1, DX (DR) est configuré comme un port de sortie.
DX (DR) DATOUT	DX (DR) data output	DX (DR) est la sortie de données lorsqu'il est configuré comme sortie.
DX (DR) DATIN	DX (DR) data input Data	DX (DR) est l'entrée de données lorsqu'il est configuré comme entrée.
FSX (FSR) FUNC	FSX (FSR) function	Contrôle la fonction de FSX (FSR). Si FSX (FSR) FUNC = 0, FSX (FSR) est configuré comme un port d'entrée/sortie. Si FSX (FSR) FUNC = 1, FSX (FSR) est configuré comme un port série.
FSX (FSR) I/O	FSX (FSR) input/output mode	Si FSX (FSR) I/O = 0, FSX (FSR) est configuré comme un port d'entrée. Si FSX (FSR) I/O = 1, FSX (FSR) est configuré comme un port de sortie.
FSX (FSR) DATOUT	FSX (FSR) data output	FSX (FSR) est la sortie de données lorsqu'il est configuré comme sortie.
FSX (FSR) DATIN	FSX (FSR) data input	FSX (FSR) est l'entrée de données lorsqu'il est configuré comme entrée.

Tableau (3.8) description des registres de contrôle des broches FSX/DX/CLKX FSR/DR/CLKR

• Le registre de contrôle du timer en réception/transmission

Ce registre de contrôle les fonctions du timer. Pendant la phase d'initialisation tout les bits sont à zéro.

La figure (3.12) illustre ce registre. Les bits 0-5 contrôle le timer de transmission, les bits 6-15 contrôle le timer de réception. Le timer du port série est similaire au timer étudié précédemment.

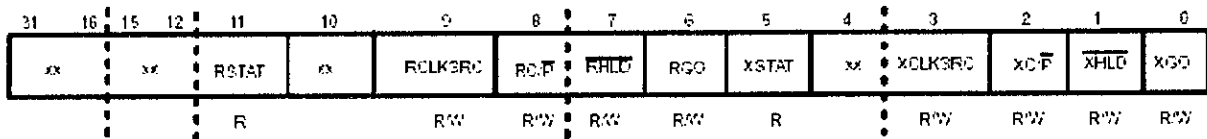


Figure (3.13) Le registre de contrôle du timer de réception/transmission [10]

Abréviation	Nom	Description
XGO	Transmit timer counter restart	Réinitialise le compteur du timer en transmission. Si XGO = 1 et le timer est activé, le compteur est mis à zéro et commence de s'incrémenter. XGO = 0 n'a aucun effet.
XHLD	Transmit counter hold signal	Si $\overline{XHLD} = 0$, le compteur en émission est désactivé. Si $\overline{XHLD} = 1$, le diviseur du signal d'horloge en émission est désactivé
XC/P	Transmit clock/pulse mode control	Lorsque XC/P = 1, le mode horloge est choisi, la sortie a un rapport cyclique de 50%. Lorsque XC/P = 0, the status flag and external output are active for one CLKOUT cycle during each timer period.
XCLKSRC	Transmit clock source	Ce bit spécifie la source d'horloge de l'émission. Lorsque XCLKSRC = 1, la source interne est utilisée pour incrémenter le compteur, elle est égale à la fréquence d'horloge du DSP divisé par 2. Lorsque XCLKSRC = 0, un signal d'horloge externe du CLKX peut être utilisé pour incrémenter le compteur.
XTSTAT	Transmit timer status	Ce bit indique l'état du timer de transmission. Une écriture sur ce bit n'a aucun effet.
RGO	Receive timer counter restart	Réinitialise et le compteur du timer en réception. Si RGO = 1 et le timer est activé, le compteur est mis à zéro et commence de s'incrémenter. RGO = 0 n'a aucun effet.
RHLD		Si $\overline{RHLD} = 0$, le compteur en réception est désactivé. Si $\overline{RHLD} = 1$, le diviseur du signal d'horloge en réception est désactivé.
RC/P	Rclock/pulse mode control	Lorsque RC/P = 1, le mode horloge est choisi. Dans le cas contraire. Le mode impulsion est choisi.

RCLKSRC	Receive timer clock source	Ce bit spécifie la source d'horloge de la réception. Lorsque RCLKSRC = 1, la source interne est utilisée pour incrémenter le compteur, elle est égale à la fréquence d'horloge du DSP divisé par 2. Lorsque RCLKSRC = 0, un signal d'horloge externe du CLKR peut être utilisé pour incrémenter le compteur.
RTSTAT	Receive timer status	Ce bit indique l'état du timer de réception. Une écriture sur ce bit n'a aucun effet.

Tableau (3.14) Description du registre de contrôle du timer de réception/transmission

- **Le registre de comptage du timer transmission/réception**

Ce registre 32 bits est divisé en deux parties : une pour la transmission (b₀-b₁₅) et autre pour la réception (b₁₆-b₃₁). Chaque compteur est remis à zéro chaque fois qu'il atteint la valeur du registre de période.

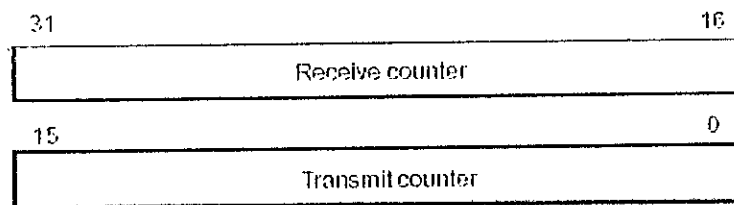


Figure (3.14) Le registre de comptage du timer transmission/réception [10]

- **Le registre période du timer transmission/réception**

Ce registre permet de spécifier la fréquence timer en transmission et en réception.

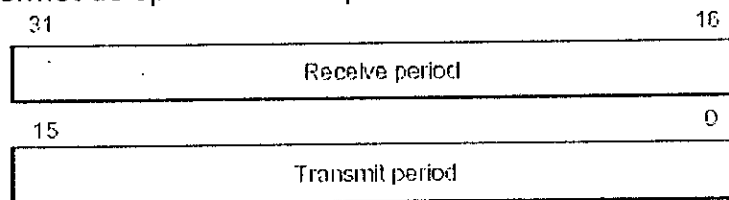


Figure (3.15) Le registre période du timer transmission/réception [10]

- **Le registre de transmission de données (DXR) (data-transmit register)**

Ce registre de 32 bits contient la donnée à transmettre. Une transition de 0 à 1 sur le bit XDRY indique que le registre DXR a été copié dans le registre série XSR et que l'on peut charger un nouveau mot.

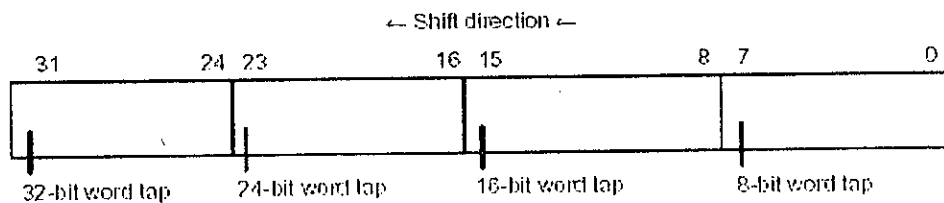


Figure (3.16) le registre de transmission de données [10]

- Le registre de réception de données (DRR) (data receive register)

Ce registre de 32 bits contient la donnée reçue. Une transition de 0 à 1 sur le bit RDRY indique que le registre DRR a été copié dans le registre série RSR et qu'il est possible de lire les données.

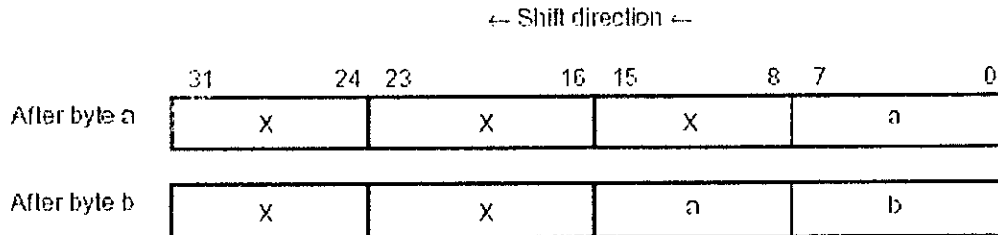


Figure (3.17) Le registre de réception de données [10]

4.6) La carte mémoire du DSP TMS320C31

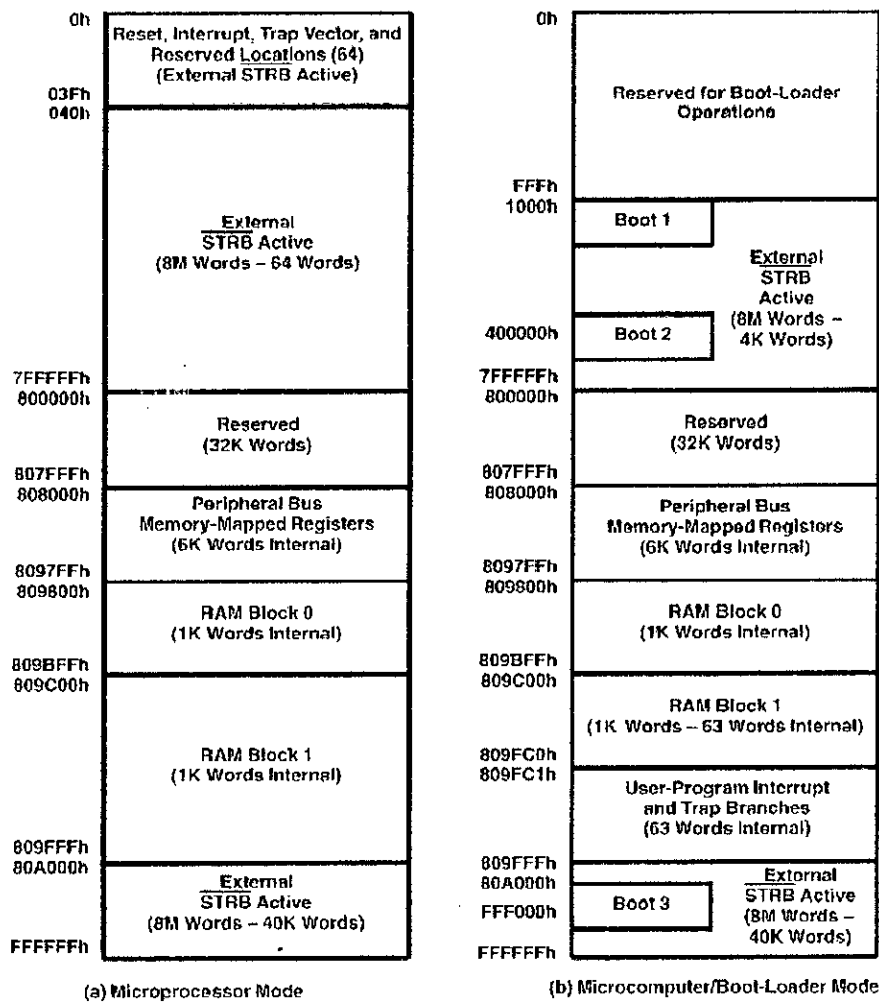


Figure (3.18) La carte mémoire du TMS320C31

808000h	DMA Global Control
808004h	DMA Source Address
808006h	DMA Destination Address
808008h	DMA Transfer Counter
808020h	Timer 0 Global Control
808024h	Timer 0 Counter
808028h	Timer 0 Period Register
808030h	Timer 1 Global Control
808034h	Timer 1 Counter
808038h	Timer 1 Period Register
808040h	Serial Global Control
808042h	FSX/DX/CLKX Serial Port Control
808043h	FSR/DR/CLKR Serial Port Control
808044h	Serial R/X Timer Control
808045h	Serial R/X Timer Counter
808046h	Serial R/X Timer Period Register
808048h	Data-Transmit
80804Ch	Data-Receive
808064h	Primary-Bus Control

Figure (3.19) La carte mémoire du timer et du port série [10]

5) Les modes d'adressage

Le mode d'adressage définit la manière de lire ou d'écrire une donnée dans la mémoire du processeur. Le DSP TMS320C31 fournit sept modes d'adressage différents qui sont :

5.1) Adressage par registres

Un registre CPU contient l'opérande.

Exemples :

ABSF R1 la valeur absolue en virgule flottante du registre R1
 FIX R0 le registre R0, de format flottant, est converti en format fixe.

5.2) Adressage direct

Dans cet adressage on donne l'adresse de l'opérande. Cette adresse sera formée par la concaténation des 8 LSBs du registre de page DP avec 16 LSBs spécifiés par le mot d'instruction (expression).

L'espace mémoire adressable maximale sans la modification du DP est de 64 K mots de 16 bits.

Syntaxe assembleur : @expression. Et l'Adresse = (8 LSBs de DP) \cup (16 LSBs de l'expression).

Exemple : ADDI @0B1DECh,R5 ; la valeur qui trouve à l'adresse 1DECh dans la page spécifiée par le registre de page DP, est ajoutée à celle contenue dans R5.

5.3) Adressage immédiat

Dans cet adressage on donne directement l'opérande (un entier ou en virgule flottante) qui est sur 16 ou 24 bits. Selon l'instruction.

Exemples : LDF 8.13,R3 : chargement de R3 par la valeur 8.13 en virgule flottante.
LDI 5,R1 : chargement de R1 par la valeur 5 en entier.

5.4) Adressage relatif PC

Cet adressage est utilisé dans les branchements étiquette ou adresse. La valeur du PC est modifiée par déplacement pour avoir l'adresse de branchement :

$$PC = PC + \text{déplacement} + 1$$

Exemple : BU ET : si PC = 2000h et ET = 2007h alors déplacement = 6.

5.5) Adressage indirect

C'est de loin le mode d'adressage le plus puissant.

- L'adresse est le contenu d'un registre ARn.
- On peut utiliser des pré ou post incrémentations / déplacements sur 8 Bits.
- On peut utiliser des registres d'index IR0/IR1.

Voici les différentes possibilités, le calcul d'adresses et leur fonctionnement :

5.5.1) Adressage indirect avec déplacement

- *ARn(dip) : Avec ajout d'un pré-déplacement (Adr = ARn + dip).
- *ARn(dip) : Avec soustraction d'un pré-déplacement (Adr = ARn - dip).
- *++ARn(dip) : Avec ajout d'un pré-déplacement (Adr = ARn + dip) et modification (ARn = ARn + dip).
- *-ARn(dip) : avec soustraction d'un pré-déplacement (Adr = ARn - dip) et modification (ARn = ARn - dip).
- *ARn ++ (dip) : Avec ajout d'un post-déplacement (Adr = ARn) et modification (ARn = ARn + dip).

- ***ARn -- (dip)** : avec soustraction d'un post-déplacement ($Adr = ARn$) et modification ($ARn = ARn - dip$) .
- ***ARn ++ (dip) %** : Avec ajout d'un post-déplacement ($Adr = ARn$) et modification circulaire ($ARn = circ(ARn + dip)$) .
- ***ARn -- (dip) %** : avec soustraction d'un post-déplacement ($Adr = ARn$) et modification circulaire ($ARn = circ(ARn - dip)$) .

5.5.2) Adressage indirect avec registres d'index

- ***+ARn(IR)** : Avec ajout d'un pré-Index ($Adr = ARn + IR$) .
- ***-ARn(IR)** : Avec soustraction d'un pré- Index ($Adr = ARn - IR$)
- ***++ARn(IR)** : Avec ajout d'un pré-Index ($Adr = ARn + IR$) et modification ($ARn = ARn + IR$) .
- ***--ARn(IR)** : avec soustraction d'un pré-Index ($Adr = ARn - IR$) et modification ($ARn = ARn - IR$)
- ***ARn ++ (IR)** : Avec ajout d'un post-Index ($Adr = ARn$) et modification ($ARn = ARn + IR$) .
- ***ARn -- (IR)** : avec soustraction d'un post-Index ($Adr = ARn$) et modification ($ARn = ARn - IR$) .
- ***ARn ++ (IR) %** : Avec ajout d'un post-Index ($Adr = ARn$) et modification circulaire ($ARn = circ(ARn + IR)$) .
- ***ARn -- (IR) %** : avec soustraction d'un post-Index ($Adr = ARn$) et modification circulaire ($ARn = circ(ARn - IR)$) .

5.5.3) adressage indirect (cas spéciaux)

- ***ARn** : ($Adr = ARn$) .
- ***ARn ++ (IR)B** : avec ajout d'un post-Index ($Adr = ARn$) et modification circulaire ($ARn = B(ARn + IR)$) .

Adr : Adresse mémoire.

dip : déplacement.

ARn : registres auxiliaires AR0 – AR7.

IR : registres d'index IR0 – IR1.

B() : Adressage par inversion de bits.

circ() : adressage circulaire.

B : utilisation de l'adressage par inversion de bits.

% : utilisation de l'adressage circulaire.

5.6) Adressage circulaire

L'adressage circulaire fait partie de l'adressage indirect sauf que le calcul d'adresse se d'une façon particulière :

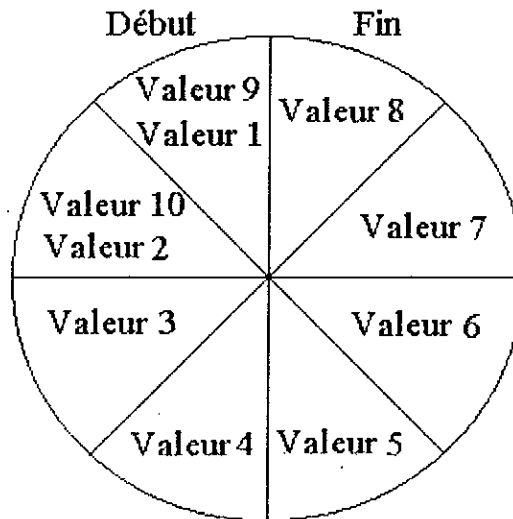
Adr = ARn

$Arn = circ(Arn \pm dip/ IR)$

Avec; $CIRC(A) = A \text{ mode}[BK]$ ou BK est le registre taille de bloc.

Principe :

Supposent un bloc de 8 positions mémoire ($BK = 8$).
On veut mettre 10 valeurs dans ce bloc par un adressage circulaire :



Figure(3.20) principe de l'adressage circulaire [16]

Règles à respecter :

On doit spécifier la taille du buffer (bloc) R dans le registre BK ($R \leq 64 K$ (une page)). Soit K le nombre de bits variant dans les adresses des positions mémoire du buffer, l'adresse de début du buffer doit contenir K zéros :

$Adrdebut = xxxxx \underbrace{000\dots 00}_{Kzéros}$; il faut que : $dip \leq R$ et $IR \leq BK$.

Les applications de ce mode d'adressage est le Filtrage numérique.

5.7) Adressage par inversion de bits

ce mode d'adressage est spécialement dédié à la FFT, il suit les même règle que l'adressage circulaire ($R \leq 64 K$, l'adresse de début du buffer doit contenir K zéros) en plus de quelques règles supplémentaires :

La table doit contenir 2^n valeur (FFT).

Si on utilise deux tables différentes pour les parties réelles et imaginaires :

$R = 2^n$; $IR0 = 2^{n+1}$; $K = n$

Si on utilise même tables consécutivement pour les parties réelles et imaginaires (Re0, Im0, Re1, Im1...etc) :

$R = 2^{n+1}$; $IR0 = 2^n$; $K = n+1$

$B(ARn + IR0) \Leftrightarrow KLSBsARn = KLSBsARn + IR0$ avec propagation de retenu en inverse (de gauche à droite).

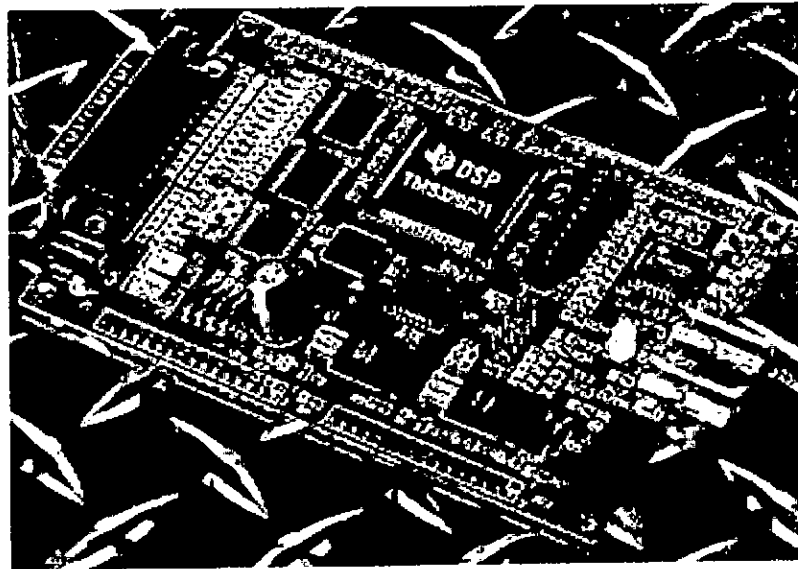
Exemple : 8 positions mémoire

$R = 8$ ($K = 3$) ; $IR0 = 100_2$ et $AR0 = xx..x000$;

- *AR0 ++ (IR0)B ; AR0 = xx..x000 (valeur 0)
 - *AR0 ++ (IR0)B ; AR0 = xx..x100 (valeur 4)
 - *AR0 ++ (IR0)B ; AR0 = xx..x010 (valeur 2)
 - *AR0 ++ (IR0)B ; AR0 = xx..x110 (valeur 6)
 - *AR0 ++ (IR0)B ; AR0 = xx..x001 (valeur 1)
 - *AR0 ++ (IR0)B ; AR0 = xx..x101 (valeur 5)
 - *AR0 ++ (IR0)B ; AR0 = xx..x011 (valeur 3)
 - *AR0 ++ (IR0)B ; AR0 = xx..x111 (valeur 7)
 - *AR0 ++ (IR0)B ; AR0 = xx..x000 (valeur 0)
 - *AR0 ++ (IR0)B ; AR0 = xx..x100 (valeur 4)
- ...etc.

6) Conclusion

Les processeurs de traitement du signal sont assez semblables aux microprocesseurs à usages générales. Ils se caractérisent par le fait qu'ils intègrent un ensemble de fonctions spéciales (jeu d'instruction, modes d'adressage, périphériques...etc). Ces fonctions sont destinées à les rendre particulièrement performants dans le domaine du traitement numérique du signal.



DSP Starter Kit

IV) Etude du TMS320C31 DSP Starter Kit.....	48
1) Introduction.....	49
2) Description générale.....	49
3) Diagramme bloc du TMS320C31 DSK.....	49
4) L'interface analogique AIC (Analogue Interface Controller)	49
4.1) Diagramme bloc de l'AIC.....	50
4.2) Brochage du circuit.....	51
4.3) Utilisation du TLC32040.....	52
a) initialisation de l'AIC.....	52
b) initialisation du Timer.....	52
c) initialisation du Port Série.....	52
d) programmation de l'AIC.....	53
• Communications primaires.....	53
• Communications secondaires.....	54
• signification des bits du registre de contrôle.....	54
5) Conclusion.....	55

1) Introduction

Dans le chapitre précédent, nous avons étudié le DSP TMS320C31 ; nous avons parlé de son architecture interne et de ses modes d'adressage. Dans ce chapitre nous ferons une description générale de la carte DSP Starter Kit qui va nous servir à implémenter notre algorithme.

2) Description générale

Le TMS320C31 DSP Starter Kit est une carte qui comporte les composants suivants :

- un processeur spécialisé TMS320C31 cadencé à 50 MHz,
- Un port parallèle qui permet la communication entre le PC et la carte,
- Un convertisseur TLC32040 à 14 bits de résolution qui intègre un filtre de reconstitution à capacités commutées,
- Des connecteurs standards RCA pour les entrées/sorties analogiques,
- Un port XDS510 pour l'émulation.

3) Diagramme bloc du TMS320C31 DSK

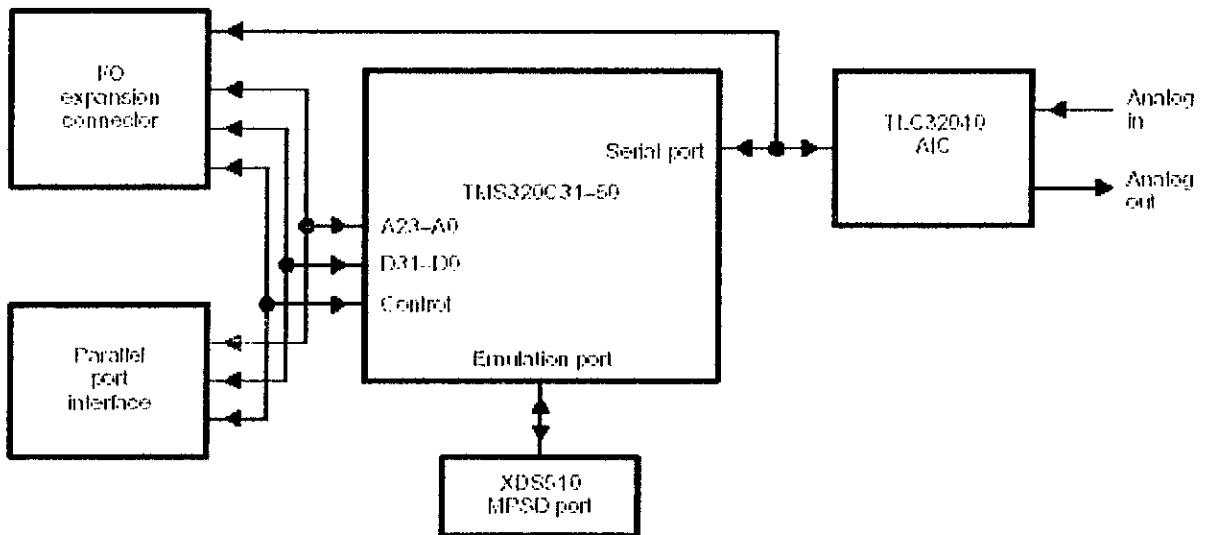


Figure (4.1) Diagramme bloc du TMS320C31 DSK

4) L'interface analogique AIC (Analogue Interface Controller)

L'interface analogique TLC32040 fait partie de la famille TLC3204X des convertisseurs analogique/numérique et numérique/analogique de qualité audio. Il se présente dans un boîtier de 28 broches (fig 4.2).

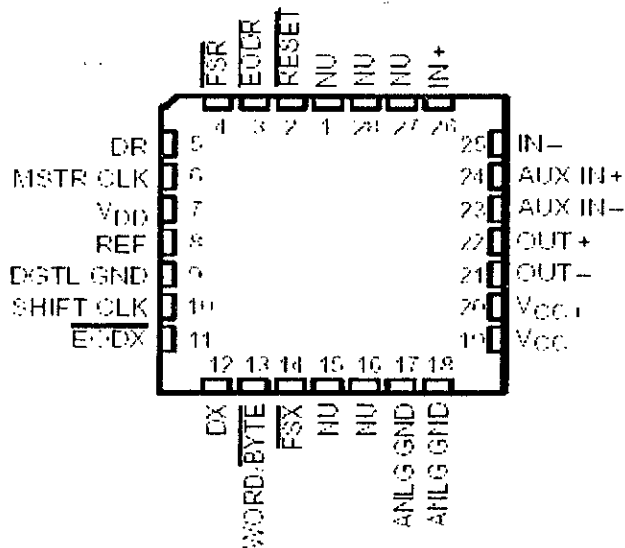


Figure (4.2) l'interface analogique TLC32040 (vue de dessus).

4.1) Diagramme bloc de l'AIC

Le TLC32040 contient les circuits suivants :

- un filtre passe bande à capacité commutée,
- un convertisseur analogique numérique à 14 bits de résolution,
- un port série interfacé directement avec quatre processeurs : TMS32011, TMS320C17, TMS32020 et TMS320C25
- un convertisseur numérique analogique à 14 bits de résolution,
- un filtre passe bas à capacité commutée.

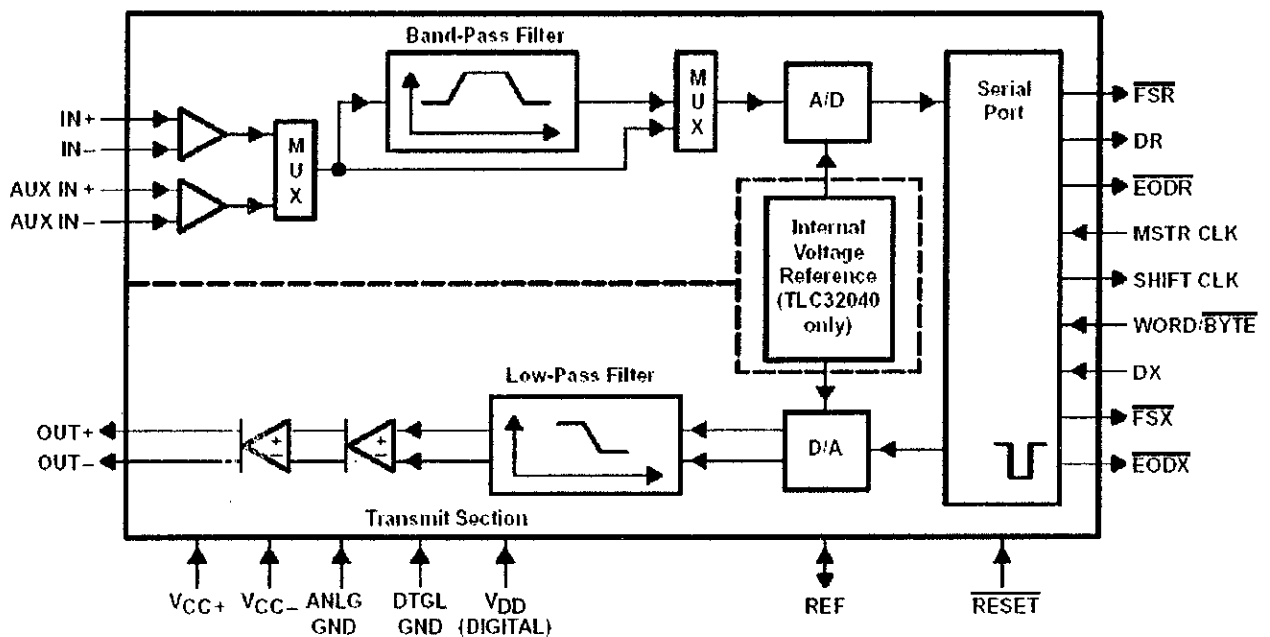


Figure (4.3) Diagramme bloc de l'AIC TLC32040

4.2) Brochage du circuit

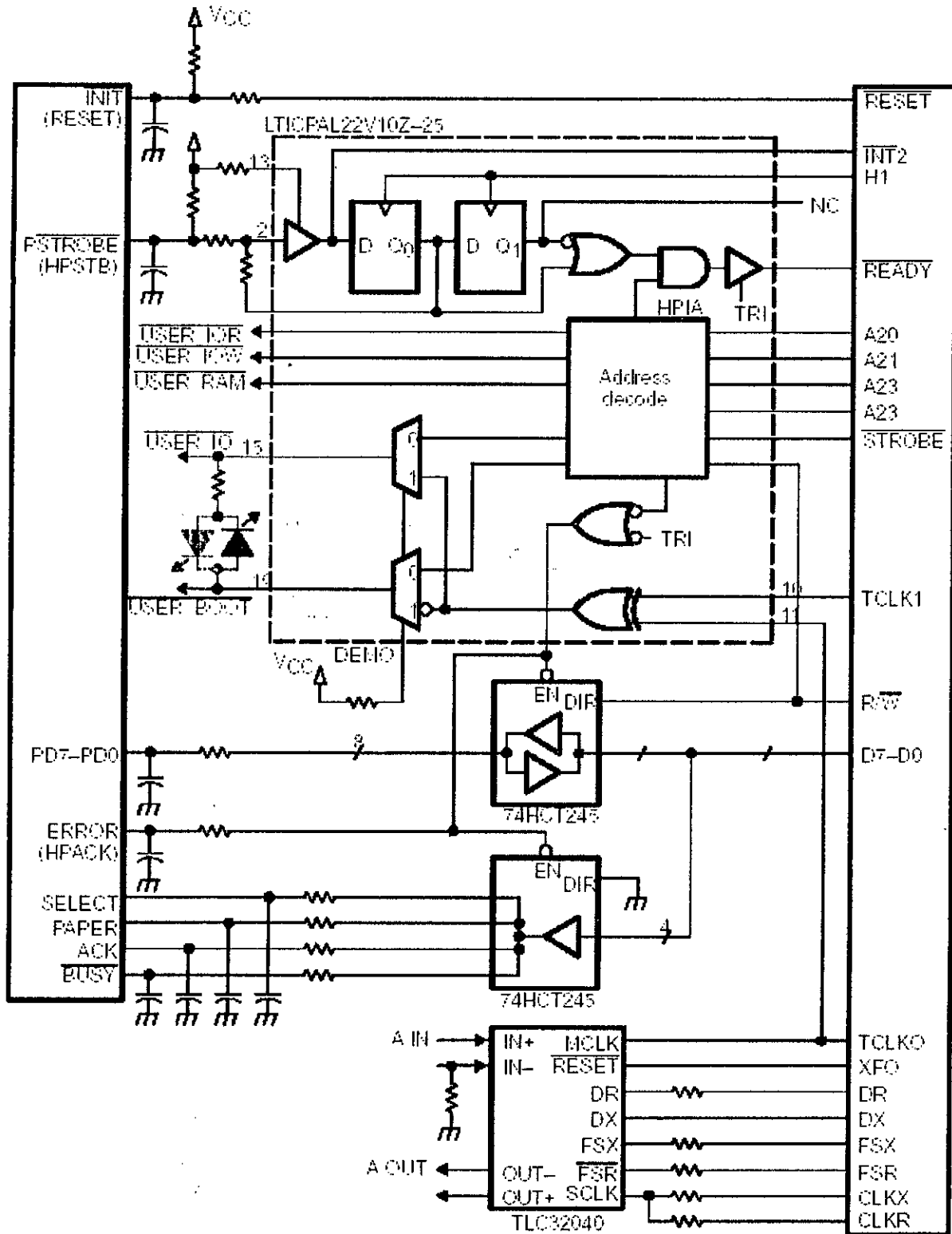


Figure (4.4) brochage du circuit

4.3) Utilisation du TLC32040

Pour pouvoir utiliser le TLC32040, il faut suivre les étapes suivantes :

a) initialisation de l'AIC

La broche XF0 du C31 est connectée à celle du $\overline{\text{RÉSET}}$ de l'AIC (Fig 4.4). Le C31 peut initialiser l'AIC en exécutant les instructions suivantes :

```
rpts 40
ldi 2h,IOF
ldi 2h,IOF
```

b) Initialisation du Timer

La broche TCLK0 du Timer est connectée à celle du MCLK (Master CLK) de l'AIC, cette horloge pilote tous les signaux de l'AIC.

Le programme ci-dessous initialise et programme le Timer pour générer un signal de fréquence égale à 6,25 MHz

```
TGCR0 .set 808020h ;   Timer 0 global control register
TCNT0 .set 808024h ;   Timer 0 counter register
TPRO  .set 808028h ;   Timer 0 period register
TIMVAL .word 3c1h ;   Timer global control register value
ldp @TGCR0 ;          Set Data Page
ldi 0h,R4 ;           Initialize R4 to zero
ldi 1h,R0 ;           Initialize R0 to 1
sti R4,@TGCR0 ;      Reset timer0
sti R0,@TPRO ;       Store timer0 period
sti R4,@TCNT0 ;      Reset timer0 counter
ldi @TIMVAL,R7 ;     Load timer control value
sti R7,@TGCR0 ;      Start timer 0
```

c) Initialisation du Port Série

pour initialiser le port série, il faut procéder comme suit :

- initialiser le port série
- autoriser les interruption en émission et en réception,
- choisir une longueur du mot égale à 16 bits en émission et en réception,
- les signaux FSX, et FSR, CLKX, CLKR sont activés au front descendant,
- sélectionner le mode continue,
- sélectionner le mode variable de transfert.

Le programme ci-dessous initialise le port série.

```
SGCR0 .set 808040h ;   Serial port 0 global control register
SPCX0 .set 808042h ;   Serial port 0 FSX/DX/CLKX control reg.
```

```

SPCR0 .set 808043h ; Serial port 0 FSR/DR/CLKR control reg.
SINIT0 .word 0e973300h; Enable RINT & 16-bit transfers
SINIT1 .word 111h ; Configure as serial port pins
ldp @SGCR0 ; Set Data Page

ldi 0h,R4 ; Initialize R4 to zero
sti R4,@SGCR0
ldi @SINIT1,R7 ; Reset and
sti R7,@SPCX0 ; initialize serial port
sti R7,@SPCR0 ; initialize serial port
ldi @SINIT0,R7 ; Reset and
sti R7,@SGCR0 ; initialize serial port
    
```

d) programmation de l'AIC

L'AIC peut être programmé en choisissant le fréquence de coupure des filtres ainsi que la fréquence d'échantillonnage. Six registres permettent ce choix qui sont:

- les registres TA,TA',TB pour la partie transmission.
- les registres RA,RA',RB pour la partie réception.

Les fréquences de conversion sont données par les relations (4.1) et (4.2)

$$f(A/D) = \frac{MCLK}{2.TA.TB} \tag{4.1}$$

$$f(D/A) = \frac{MCLK}{2.RA.RB} \tag{4.2}$$

Les fréquences de coupure des filtres sont données par les relations (4.3) et (4.4)

$$f(A/D) = \frac{MCLK}{2.TA} \tag{4.3}$$

$$f(D/A) = \frac{MCLK}{2.RA} \tag{4.4}$$

• **Communications primaires**

Le format de la donnée dans une communication primaire est indiqué dans la figure ci-dessous :

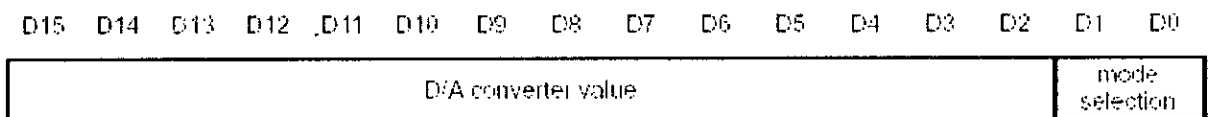


Figure (4.5) format de la donnée dans une communication primaire

La donnée est sur 16 bits ; les 14 MSBs sont réservés aux données converties par l'AIC, et le reste (D0-D1) est réservée aux modes de sélection.

L'AIC envoie la valeur de la donnée vers le convertisseur A/N et active l'un des modes de sélection suivants

LSBs	Mode de sélection
00	Tx Counter A ← TA, Rx Counter A ← RA Tx Counter B ← TB, Rx Counter B ← RB
01	Tx Counter A ← TA + TA', Rx Counter A ← RA + RA' Tx Counter B ← TB, Rx Counter B ← RB
10	Tx Counter A ← TA - TA', Rx Counter A ← RA + RA' Tx Counter B ← TB, Rx Counter B ← RB
11	Tx Counter A ← TA, Rx Counter A ← RA Tx Counter B ← TB, Rx Counter B ← RB

Tableau (4.1) les modes de selection de la communication primaire

Le deuxième et le troisième modes utilisent les registres TA' et RA' pour accélérer ou ralentir la fréquence d'échantillonnage

- Communications secondaires**

La communication secondaire suit la communication primaire qui consiste à choisir un mode de sélection. la communication secondaire programme l'AIC par le chargement des registres A, A',B et le registre de contrôle. La figure ci-dessous illustre le format de la donnée dans une communication secondaire.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	TA register value (unsigned)					X	X	RA register value (unsigned)					0	0
X	TA' register value (signed 2s complement)					X	RA' register value (signed 2s complement)					0	1		
X	TB register value (unsigned)					X	RB register value (unsigned)					1	0		
X	X	X	X	X	X	X	X	Control register					1	1	

Figure (4.6) format de la donnée dans une communication secondaire

- signification des bits du registre de contrôle**

Le registre de contrôle possède six bits dont la signification de chaque bit est

donnée dans la figure ci-dessous

D7	D6	D5	D4	D3	D2
Input Gain		Transmit/Receive	AUX IN Pins	Loopback Function	Bandpass Filter
0 0 = 1X for \pm 6V analog input		0 = asynchronous	0 = disables	0 = disables	0 = deletes
0 1 = 2X for \pm 3V analog input		1 = enables	1 = enables	1 = enables	1 = inserts
1 0 = 4X for \pm 1.5V analog input					
1 1 = 1X for \pm 6V analog input					

Tableau (4.2) signification des bits du registre de contrôle

5) Conclusion

Dans ce chapitre, on a fait une description du TMS320C31 DSK. Ce kit de développement dispose d'un DSP TMS320C31 qui très utilisé en communication. L'interface analogique AIC permet la communication entre la carte et les dispositifs externes (microphone, haut parleur..etc). Pour pouvoir utiliser ce circuit (AIC), il faut l'initialiser avec timer et le port série, ensuite le programmer.

V) Mise en œuvre du codeur.....	56
1) introduction.....	57
2) Description du programme.....	57
2.1) L'initialisation des périphériques.....	57
2.2) Les entrées/sorties.....	57
2.3) Analyse du signal vocal.....	57
2.4) Codage et décodage des paramètres LSF _i	57
2.5) synthèse du signal vocal.....	58
2.6) Evaluation du codeur.....	59
3) Conclusion.....	59

1) Introduction

Dans le chapitre précédent, nous avons fait une description générale du TMS320C31 DSP Starter Kit. Dans ce chapitre nous ferons une description de notre programme ensuite nous discuterons les résultats obtenus.

2) Description du programme

Notre programme est écrit en langage C, il comprend le codeur et le décodeur et est divisé en cinq grande parties.

2.1) L'initialisation des périphériques

Routines	Description
init_periph	Initialise les périphériques (le timer et le port série).
Init_aic	Initialise l'AIC.
Prog_aic	Programme l'AIC.

Tableau (5.1) les routines d'initialisations des périphériques

2.2) Les entrées/sorties

Routines	Description
input	Lit l'entrée de l'AIC.
output	Prépare la transmission de données à travers c_int05.
C_int05	Interruption du port série en transmission.
C_int06	Interruption du port série en réception.

Tableau (5.2) les routines d'entrées/sorties

2.3) Analyse du signal vocal

Routines	Description
preemphasis	Pré-accentue le signal vocal.
Hamming	Calcule de fenêtre de Hamming.
residu	Calcule l'erreur de prédiction (résidu).
autocorrelation	Calcule de la matrice d'autocorrélations.
wld	Inverse la matrice d'autocorrélations.
chebps	Evalue les séries des polynômes de Chebishev.
ptolsfnd	calcule les LSF à partir des coefficients de prédiction.

Tableau (5.3) les routines d'analyse

2.4) Codage et décodage des paramètres LSF_i

Notre programme utilise une seule routine pour coder et décoder les paramètres LSF_i. Cette routine est appelée "search_table", elle permet de chercher dans un dictionnaire (placé en émission et en réception) le paramètre le plus proche du LSF calculé lors de l'analyse du signal vocal. Seule l'indice du dictionnaire est à transmettre. A la réception, le LSF est directement déterminé à partir de son indice.

2.5) synthèse du signal vocal

Routines	Description
get_lsfpol	trouve les polynômes $F_1(z)$ ou $F_2(z)$ à partir des LSF_i .
lsftop	transforme les LSF_i en coefficients de prédiction.
Syn_filt	Reconstitue le signal vocal à partir des coefficients a_i et du résidu.

Tableau (5.4) les routines de synthèse

La figure (2.1) donne une représentation complète de notre programme sous forme d'organigramme.

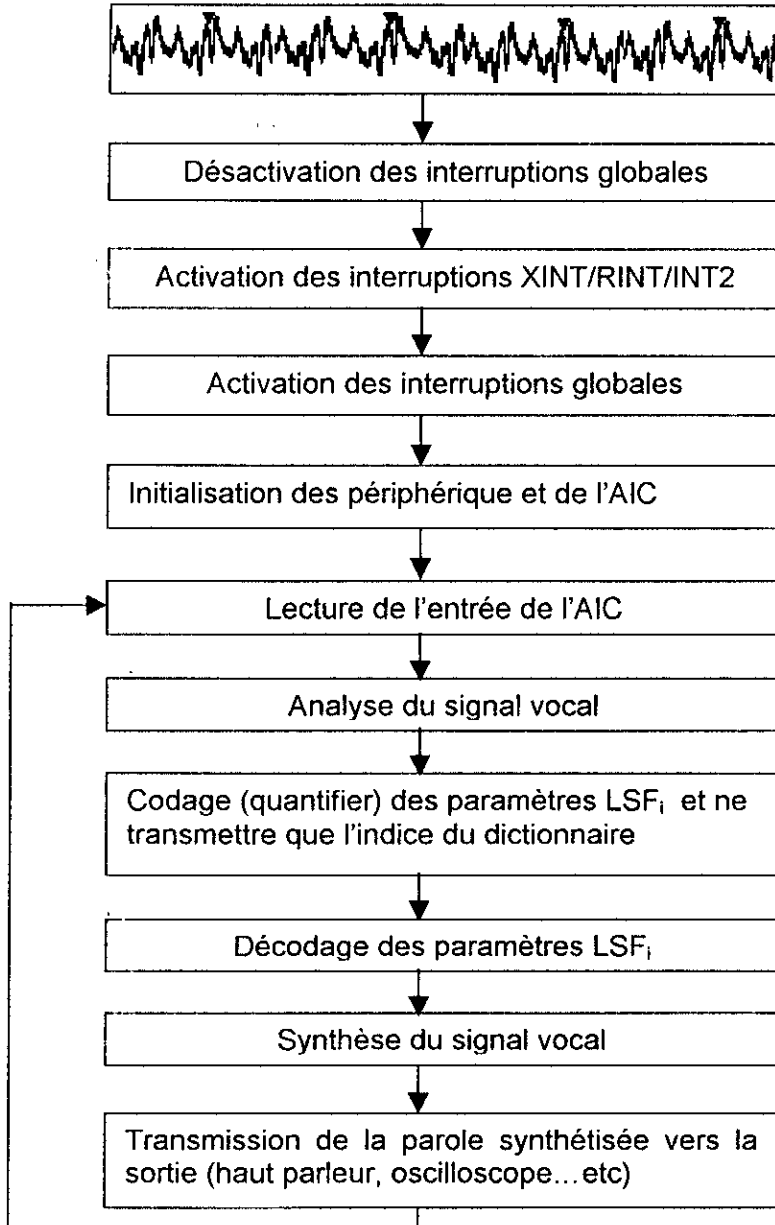


Figure (5.1) Organigramme du codeur LPC

3) Evaluation du codeur

pour évaluer notre codeur, on visualise deux signaux originaux et deux signaux synthétisés, et on les compares. Le logiciel utilisé pour la visualisation est appelé "VISU".

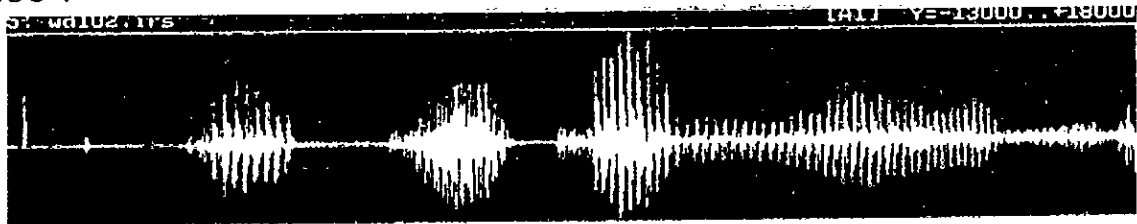


Figure (5.2) signal original n°1



Figure (5.3) signal synthétisé n°1.

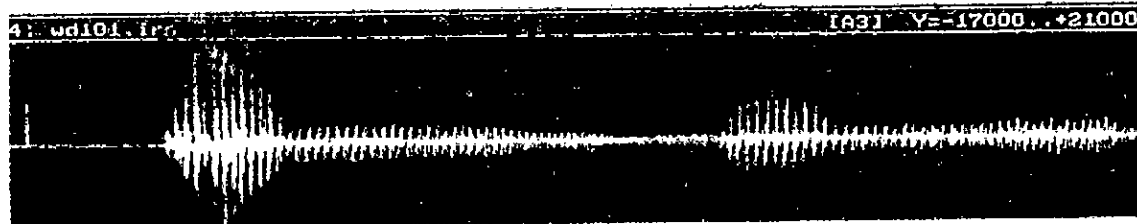


Figure (5.4) signal original n°2.

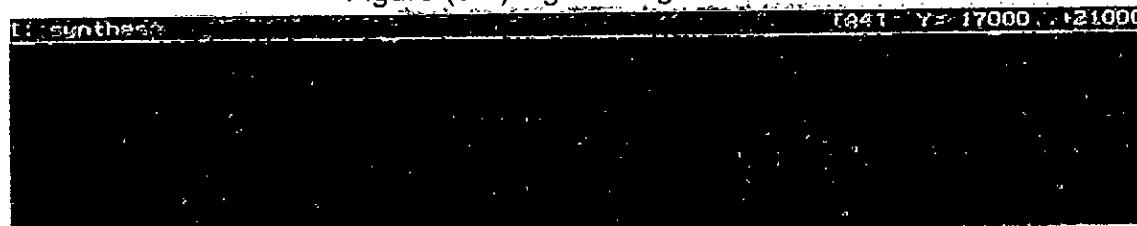


Figure (5.5) signal synthétisé n°2.

On remarque que les deux signaux sont similaires, mais il y a un petit décalage en temps, ce décalage est dû au temps du codage.

4) Conclusion

Dans ce chapitre, on a pu décrire les parties les plus importantes de la mise en œuvre de notre application. On a pu présenter quelques exemples de tests et évaluer les résultats obtenus.

Conclusions générales

Dans ce travail, on a étudié et implémenté une partie du codeur LPC sur une carte DSP TMS320C31.

Ce travail m'a permis, par l'étude du codage prédictif linéaire, de comprendre le principe du modèle de production de la parole, la prédiction linéaire ainsi que le codage LPC.

Dans ce type de codage, on code certains paramètres du signal décomposé, contrairement au codage de formes d'ondes (wave forms coding), où on code le signal.

Ce travail, m'a permis aussi, par l'étude du kit DSK, de comprendre l'architecture du DSP TMS320C31, ses modes d'adressage et les outils de développement utilisés.

La qualité de la parole synthétisée est assez proche de celle d'origine.

On souhaite que le travail réalisé soit suivi par une implémentation de l'algorithme LPC sur une carte FPGA.

Bibliographie

- [1] M.Kunt, R.Boite "traitement de la parole", Presses Polytechniques Et Romandes, 1987.
- [2] J.Makhoul "Linear Prediction : A Tutorial Review," *proceeding of the IEEE*, VOL 63, NO 4, April 1975.
- [3] R.Boite, H.Bourlard, T.Dutoit, J.Hancq et H.Leich "Traitement de la parole", Presses Polytechniques Et Universitaires Romandes,2000.
- [4] F.FLITTI, " Codeur harmonique à bas débit ", thèse de Magister, E.N.P 1999.
- [5] M.Djeddou, "Conception et réalisation d'un codeur-décodeur de la parole à bande étroite (300 3400 Hz) à 16 kbits/s et à faible retard " , thèse de Magister, E.N.P 1997.
- [6] "Interpolation of linear prediction coefficients for speech coding", Master thesis, McGill university, April 2000 (IslamT2000.pdf).
- [7] F.Merazka "Quantification des paramètres LSF" thèse de magistère, E.N.P 1997.
- [8] G.Baudoin, F.Viroleau, "Les processeurs de traitement du signal famille 320C5x"-Dunod.
- [9] G.Baudoin, J.Cernocky, P.Gournay, G.Chollet "Codage de la parole à bas et très bas débit", annale de télécommunication,2000.
- [10] TMS320C3X User's guide (spru031e.pdf).
- [11] TMS320C31 Embedded Control (spru083.pdf).
- [12] TMS320C3X DSP Starter Kit User's guide (spru163a.pdf).
- [13] TMS320C3X/C4X Optimizing C Compiler User's guide (spru034h.pdf).
- [14] TLC32040 Data Sheet (slas014e.pdf).
- [15] Manual Update Sheet (sprz130.pdf).
- [16] A.Ahmed Said, "Mise en œuvre d'une carte DSP TMS320C30", projet de fin d'étude, E.N.P, juin 1999.
- [17] B.Mohamed, C.Sid-Ali "Etudes des processeurs de traitement du signal (DSP), famille TMS320C3x " Séminaire, ENP 2001.

[18] J.HERNANDEZ "Algorithmes d'acquisition, compression et restitution de la parole à vitesse variable. Etude et mise en place ", projet de fin d'études, Ecole Nationale Supérieure de l'Electronique et de ses Applications, 1995.

[19] Adaptive Differential Pulse Code Modulation (adpcm.pdf).

Calcul du Gain de prédiction

Pour synthétiser la parole, le modèle autorégressif est excité soit par une suite de bruit blanc, soit par des impulsions périodiques. On peut calculer le gain en faisant appel au critère suivant :

L'énergie totale contenue dans la réponse impulsionnelle doit être égale à l'énergie totale du signal analysé.

Le signal reconstitué est donné par la formule :

$$s(n) = - \sum_{k=1}^p a_k s(n-k) + G\delta(n) \quad (1)$$

où $\delta(n)$ est l'impulsion de Dirac.

L'équation (1) peut se mettre sous la forme :

$$\begin{cases} s(n)=0 & \text{pour } n < 0 \\ s(n)=G & \text{pour } n=0 \\ s(n) = - \sum_{k=1}^p a_k s(n-k) & \text{pour } n > 0 \end{cases} \quad (2)$$

L'autocorrélation du signal prédit est :

$$\begin{cases} \hat{R}(j) = \sum_{n=-\infty}^{+\infty} \hat{s}(n) \hat{s}(n+j) \\ \hat{R}(0) = \sum_{n=0}^{+\infty} \hat{s}(n)^2 \end{cases} \quad \text{pour } j=0 \quad (3)$$

en remplaçant $s(n)$ par sa valeur de (3) et en distinguant les cas $n=0$ et $n \geq 1$, l'expression de $R(0)$ devient :

$$\hat{R}(0) = \hat{s}(0)^2 + \sum_{k=1}^p a_k \sum_{n=k}^{+\infty} \hat{s}(n) \hat{s}(n-k) \quad (4)$$

en posant $m=n-k$

$$s(m)=0 \quad \text{pour } m < 0 \quad (5)$$

on a

$$\begin{cases} \hat{R}(0) = G^2 + \sum_{k=1}^p a_k \left(\sum_{m=1-k}^{+\infty} \hat{s}(m) \hat{s}(m+k) \right) \\ \hat{R}(0) = G^2 + \sum_{k=1}^p a_k \hat{R}(k) \end{cases} \quad (6)$$

D'où

$$G^2 = \hat{R}(0) - \sum_{k=1}^p a_k \hat{R}(k) \quad (7)$$

$$\begin{cases} \hat{R}(j) = \sum_{k=0}^{+\infty} \hat{s}(k) \hat{s}(k+j) & \text{pour } j > 1 \\ \hat{R}(j) = \sum_{k=1}^{+\infty} a_k \hat{R}(j-k) \end{cases} \quad (8)$$