

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

## **Ecole Nationale Polytechnique**



المدرسة الوطنية المتعددة التقنيات  
 Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات  
 BIBLIOTHEQUE — المكتبة  
 Ecole Nationale Polytechnique

**Département de Génie Electrique**

### **Projet de fin d'Etudes**

**Pour l'obtention du diplôme**  
**D'Ingénieur d'état en Electrotechnique**



**Mise en œuvre de la carte dSPACE DS1104**  
**Application à la génération de MLI pour la**  
**commande des Onduleurs à deux et trois niveaux**

**Etudié par :**

**M<sup>R</sup>: S.MERAZGA**

**M<sup>R</sup>: S.LOUNIS**

**Proposé et dirigé par :**

**M<sup>R</sup>: E.M.BERKOUK**

**E.N.P. 10, Avenue Hassen-Badi, El Harrach, ALGER**

**Promotion : juin 2004**

# Remerciements

Nous remercions Dieu de nous avoir donné la force pour accomplir ce travail.

A travers ce mémoire on adresse nos sincères remerciements à notre promoteur **M<sup>R</sup>: Berkouk** pour son aide, son encouragement et sa patience.

Nous remercions ainsi **M<sup>R</sup>: Benmansour** pour sa disponibilité et son aide.

Nous tenons aussi à remercier **M<sup>R</sup>: LARBESSE** qui nous a fait l'honneur de juger ce travail.

**Résumé :** Présentation général des caractéristiques techniques de notre carte ds1104, ainsi que son logiciel, et implémentations d'algorithmes de génération de signaux de commande MLI pour des onduleurs deux et trois niveaux.

Mots clés : Le DSP esclave TMS320F240, le processeur maître MPC8240, dSPACE, RTI, ControlDesk, Onduleur, la MLI.

**Summary :** General presentation of the design features of our card ds1104, like its software, and implementations of algorithms of generation of control signals PWM for inverters two and three levels.

Key words : The slave DSP TMS320F240, the master MPC8240, dSPACE, RTI, ControlDesk, inverters, the PWM.

**ملخص** عرض الخصائص التقنية للبطاقة الإلكترونية و وضع الاقورتمات التي تسمح بإنتاج أمواج السيطرة لمولد التيار المتواتر.  
مفتاح الكلمات: البروسييسور عبيد، البروسييسور ملك، دسبييس، رتي، قرص التحكم، مولد التيار المتواتر الموجة ملي.

## SOMMAIRE

Notations utilisées (Nomenclature)	1
Introduction générale	1

---

### CHAPITRE I

#### Présentation générale des DSP

---

Introduction	3
I.1.Le marché des DSP	3
I.1.1.Les différents constructeurs	3
I.1.2.Catégories des DSP	4
I.2.Format de calcul des DSP	4
I.3.Domaines d'application des DSP	5
I.4.Les familles de DSP de Texas Instruments	6
I.5.Les différences entre les DSP et les microprocesseurs	10
I.6.Architecture de l'environnement externe au DSP	12
Conclusion	13

---

### CHAPITRE II

#### Caractéristiques techniques de DSPACE DS1104

---

Introduction	15
II.1.Présentation de la carte d'acquisition DS1104 de DSPACE	15
II.2.Le processeur maître MPC8240	17
II.3.DSP esclave	21
II.3.1.Description	21
II.3.2.Brochage	22
II.3.3.Description des principales broches du TMS320F240	22
II.3.4. Architecture du TMS320F240	26
II.3.5. Cartographie mémoire	27
II.3.6.CPU	28
II.4.La mémoire interne	33
II.5.Le module d'horloge	34

II.6.Périphériques	34
II.7.Interruption et RESET	37
II.8.Mode d'adressage	38
II.9.Les dispositifs contrôlés par le processeur maître	39
Conclusion	39

---

### CHAPITRE III

#### Présentation et mise en œuvre de la carte

---

Introduction	44
III.1.Utilisation de Control Desk	44
III.2.Utilisation de RTI	56
III.3.Bibliothèque d'interface Matlab-DSPACE	61
Conclusion	62

---

### CHAPITRE IV

#### Applications

---

Introduction	63
IV.1.Stratégie de commande d'un onduleur à deux niveaux	63
IV.2.Stratégie de commande d'un onduleur à trois niveaux	67
IV.2.1.Commande triangulo sinusoïdale à une seule porteuse	68
IV.2.2.Commande triangulo sinusoïdale à deux porteuses	72
IV.3.Interprétation des résultats	75
Conclusion	75
Conclusion générale	76

---

### BIBLIOGRAPHIE

### ANNEXES :

- ANNEXE I
- ANNEXE II
- ANNEXE III



## Nomenclature

**DLL:** Delay Lock Loop.

**PLL:** (Phase Locked Loop).

**IU:** (Integer Unit).

**MMU:**(Memory Management Unit).

**FPU:** (Floating Point Unit).

**LSU:** (Load-Store Unit).

**SRU:** (System Register Unit).

**TLB:** (Translation Lookaside Buffer).

**BPU:** (Branch Processing Unit).

**LRU:** (Least Recently Used).

**I<sup>2</sup>C:** (Inter Integrate Circuit).

**DMA :** (Acces Direct Memory).

**ADC, DAC :** (Convertisseurs)

**A,B,C:** Indices correspondant aux trois phases "A", "B", "C".

**U<sub>C1</sub>:** Première tension d'alimentation continue de l'onduleur.

**U<sub>C2</sub>:** Deuxième tension d'alimentation continue de l'onduleur.

**M :** Point milieu.

**K:** Bras d'onduleur (K= 1,2,3).

**T<sub>ik</sub>, DD<sub>jk</sub> :** Indice du transistor et de la diode d'un bras (i,j,k=1,2,3 ).

**TD :** Interrupteur bidirectionnel.

**B<sub>Ks</sub> :** Signal de commande de base du transistor T<sub>Ks</sub> du bras K.

**V<sub>AM</sub> :** Tension de la phase A par rapport au point milieu M de la source de tension d'entrée.

**V<sub>AN</sub> :** Tension de la phase A par rapport au neutre de la charge triphasée équilibrée.

**V<sub>m</sub> :** Amplitude maximale de référence sinusoïdale.

**U<sub>pm</sub> :** Amplitude maximale de la porteuse.

**m :** Indice de modulation.

**r :** Taux de modulation.

**f :** Fréquence de la référence.

**f<sub>p</sub> :** Fréquence de la porteuse.

## *Introduction générale*

Depuis plus de 10 ans, la société dSPACE a été parmi les premiers fournisseurs en outils de développement, et de test de nouveaux systèmes de commande. Plus de 10000 systèmes dSPACE sont en fonction dans le monde [16].

Ses produits principaux sont surtout utilisés dans l'électronique des véhicules, mais ils peuvent aussi être utilisés à domicile, ou dans l'aéronautique, ou pour l'entraînement et l'automatisation industrielles.

Les systèmes dSPACE permettent aux contrôleurs et à l'unité de contrôle électronique (ECU) de réduire leurs temps de développement, leurs coûts d'une manière importante, et d'augmenter leurs productions. Cela est rendu possible par le mélange optimal des solutions standards pour le prototypage du contrôle rapide, la simulation du matériel en boucle et le calibrage [16].

La mise au point de nouvelles machines électriques, de convertisseurs, ... dépend largement du développement d'algorithmes de plus en plus complexes. Les solutions dSPACE s'adaptent parfaitement à ce marché transversal qui prend une place de plus en plus importante.

De nombreuses écoles et universités utilisent les produits et solutions dSPACE afin de former leurs étudiants à l'automatique et aux techniques de contrôle en se familiarisant avec les :

- Systèmes mono-cartes DS1104 et DS1103.
- Systèmes modulaires DS1005.
- Logiciels d'expérimentation : ControlDesk Standard, ControlDesk Test Automation, MLIB/MTRACE.
- Logiciels d'implémentation : RTI, RTI-MP, RTI CAN Blockset, Compilateurs.
- Logiciels de génération de code de série : TargetLink Base Suite, TargetLink Optimisation Modules, TargetLink, Target Simulation Module, TargetLink ASAP2 Module.

Ce modeste travail a pour but de mettre en œuvre la carte DSP « DS1104 de dSPACE » ainsi que son logiciel d'évaluation, de faciliter son utilisation, de monter les méthodes de programmation de cette carte et créer une plateforme de travail grâce à des routines réutilisables en simulink.

Nous commençons notre exposé par une brève introduction (chapitre I) puis nous étudierons les caractéristiques techniques de la carte dSPACE et l'architecture du TMS320F240 au chapitre II.

Ce n'est qu'au chapitre III que nous présenterons le logiciel de la carte (ControlDesk) fourni par dSPACE. Et au dernier chapitre nous montrerons comment implémenter des programmes en simulink (exemple d'une MLI calculée pour un onduleur à trois niveaux).

Ce manuel représente la synthèse de tous ce qu'il faut savoir pour utiliser le kit de développement, mais néanmoins, il ne peut remplacer la documentation officielle et détaillée de dSPACE dont il faut toujours se référer.

Le contenu de ce mémoire "mise en œuvre d'une carte DSP" est tiré d'un travail de synthèse qu'on a effectué dans le cadre de notre projet de fin d'études à l'E.N.P.

Ce texte est maintenu de nombreux exemples, images et documents, issus des documentations des constructeurs de DSP et de professionnels du domaine. A chaque fois que le cas se présente, vous y trouverez la référence exacte de la source (auteur) des documents présentés (livres, articles), voire un lien internet (sites des constructeurs) quand c'est possible.

On encourage vivement le lecteur à approfondir sa réflexion en profitant de ces références / liens.

Lors de la rédaction, On a aperçu qu'il est très difficile de trouver des textes d'introductions aux DSP (a priori en langue française). Le web et les librairies techniques ou universitaires renferment énormément de documents intéressants mais pointus, fait par des spécialistes pour des spécialistes. Ceci est peut-être dû au fait que les DSP sont naturellement issus du monde des microprocesseurs. Domaine qu'un ingénieur en électronique ou en électrotechnique est sensé maîtriser parfaitement.

Enfin, pour le prototypage, certains constructeurs proposent des cartes à DSP directement programmable à partir d'un langage de très haut niveau ou même d'une interface graphique. Par exemple, la société dSPACE fournit des produits programmables à partir des *schémas blocs* de Simulink : un premier post-processeur traduit le schéma en C, puis le compilateur est lancé automatiquement.

# Chapitre: I

---

## Présentation générale des DSPs

## Introduction

Depuis maintenant plusieurs années, le traitement numérique du signal est une technique en plein essor. Nous allons nous intéresser aux processeurs de traitements du signal, plus communément désignés par l'acronyme Anglais DSP (Digital Signal Processor).

Au niveau historique, les DSPs ont été initialement développés pour des applications de radars militaires et de télécommunications cryptées dans les années 70. C'est Texas Instruments® qui en 1978 introduit un DSP pour la synthèse de la voix pour des applications grand public. Il aura fallu 15 ans supplémentaires pour que les DSPs deviennent des composants incontournables de l'électronique.

Les domaines d'application du traitement numérique du signal sont nombreux et variés, nous en verrons une liste ci-après. Chacun de ces domaines nécessite un système de traitement numérique, dont le coeur est un et même parfois plusieurs DSP ayant une puissance de traitement adaptée, pour un coût économique approprié.

Un DSP est un type particulier de microprocesseur. Il se caractérise par le fait qu'il intègre un ensemble de fonctions spéciales. Ces fonctions sont destinées à le rendre particulièrement performant dans le domaine du traitement numérique du signal.

Comme un microprocesseur classique, un DSP est mis en oeuvre en lui associant de la mémoire (RAM, ROM) et des périphériques à la différence qu'un DSP typique a plutôt vocation à servir dans des systèmes de traitements du signal. Ainsi il se présente généralement sous la forme d'un microcontrôleur intégrant de la mémoire, des timers, des ports séries synchrones rapides, des contrôleurs DMA, des ports d'E/S divers.

### I.1 Le marché des DSP :

#### I.1.1 Les différents constructeurs

Le marché est partagé entre quatre constructeurs : le premier est Texas Instruments suivi par Analog Device puis Motorola et enfin Lucent.

Les différences entre modèles de DSP se situent au niveau :

- Du format de calcul : fixe ou en flottant.
- De la taille du bus de donnée : 16, 24 ou 32 bits.
- De la puissance en Millions d'Instructions Par Seconde (MIPS).

Actuellement, le marché des DSP est réparti entre :

- Texas Instruments (47 %).
- ATT (35 %).
- Motorola (27 %).
- ADI (8 %).

### 1.1.2 Catégories des DSP

On peut ranger les DSP d'aujourd'hui en deux catégories (du point de vue de notre sujet):

- a) DSP généralistes (à virgule fixe ou flottante).
- b) DSP dédiés à la commande des moteurs.

Par exemple, Texas Instruments propose, dans la série TMS320F240 :

- 20 MIPS (virgule fixe 16 bits).
- 12 sorties MLI.
- 3 bases de temps.
- 2 convertisseurs AD 10 bits.
- 9 comparateurs.
- une logique intégrée de gestion des temps morts.

### 1.2 Format de calcul des DSPs

Deux familles distinctes de DSP sont proposées par les constructeurs, les processeurs à virgule fixe et les processeurs à virgule flottante.

Les processeurs à virgule fixe lisent les bits comme des fractions en puissance négative de 2. Ainsi le nombre 0.75 est traité comme  $0.5+0.25$ .

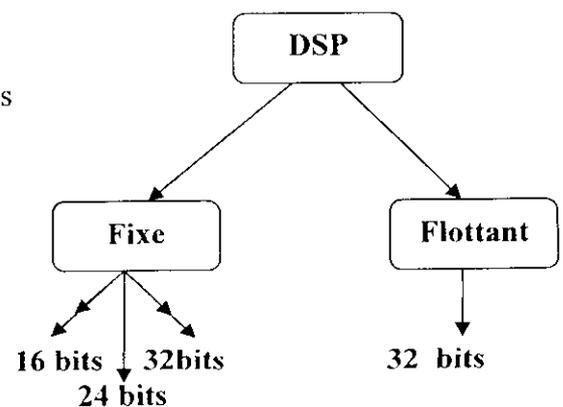


Figure I.1: Format de calcul des DSPs.

### **I.3 Domaines d'application des DSP**

A l'origine, les deux principaux domaines d'application des DSPs ont été les télécommunications et le secteur militaire. Aujourd'hui, les applications se sont diversifiées et s'orientent vers le multimédia, l'électronique grand public et l'informatique graphique mais aussi l'automatique, le domaine médical (traitement et archivage d'images, analyse de signaux électrocardiographiques, implants cochléaires...), l'instrumentation, l'électronique automobile (détection de cliquetis, système ABS...) [1].

Les DSPs sont utilisés dans des domaines très variés ; voici quelques exemples des applications les plus courantes :

#### **I.3.1 Dans le domaine des télécommunications :**

- La téléphonie filaire : les modems, les codeurs de paroles, les multiplexeurs, les télécopieurs, les minitels intelligents.
- La radiotéléphonie : téléphone cellulaire, téléphone sans fil (les codeurs de paroles GSM, les modem radio).

#### **I.3.2 Dans le domaine militaire :**

Le guidage de missiles, la navigation, les modems radio, les communications cryptées, le traitement radar et le traitement sonar.

#### **I.3.3 Dans le domaine médicale :**

- La compression d'images médicales (radiologie, échographie...) en vue de leurs archivages.
- Le traitement des signaux biophysiques.

#### **I.3.4 Dans le domaine de l'électronique automobile :**

- Les équipements de contrôle moteur.
- La détection de cliquetis pour l'optimisation de l'avance à l'allumage.
- L'aide à la navigation.
- Les commandes vocales.

#### **I.3.5 Dans le domaine de l'instrumentation :**

- Les analyseurs de spectre.
- Les générateurs de fonctions.
- L'interprétation des signaux sismiques.

### **1.3.6 Dans le domaine de l'automatisation et du contrôle des processus :**

- La surveillance et la commande des machines.
- Les robots et les servomécanismes.

Le contrôle des moteurs.

#### **❖ Utilisation du DSP dans le contrôle des moteurs :**

Le contrôle numérique des moteurs [4] est bien connu depuis longtemps. Toutefois, les contrôles sophistiqués qui réclament de grandes capacités de traitement, ne sont rendus possibles que depuis relativement peu, avec l'avènement des DSP dont l'architecture particulière est bien adaptée à l'implémentation d'algorithmes de commande [5].

Le DSP peut comporter dans son boîtier des convertisseurs analogique- numérique et numérique- analogique et plus généralement l'ensemble des ressources périphériques qu'on trouve classiquement dans les microcontrôleurs. Le DSP, dans ce cas, intègre les éléments combinés d'un puissant calculateur et d'un microcontrôleur. On parle alors de contrôleur DSP.

Dans la vaste gamme disponible, les DSP à virgule flottante peuvent être utilisés dans les applications de grande précision pour lesquels l'optimisation de coût est secondaire.

Par contre, les DSP 16 bits à virgule fixe, avec des performances allant de quelques MIPS ; sont parfaitement capables de superviser tous les aspects de la commande d'un moteur .Ces solutions, relativement économiques, sont maintenant passées dans le monde industriel.

On rappelle que Texas Instruments commercialise un module d'évaluation qui permet de développer ce type d'applications (la commande du moteur) et de valider les stratégies de contrôle avant de développer son propre hardware.

## **1.4 LES FAMILLES DE DSP DE TEXAS INSTRUMENTS**

En 1982, Texas Instruments a lancé son premier DSP (le TMS32010). Depuis, Texas s'est imposé comme le principal fabricant de processeurs de traitement de signal. La famille de DSP TMS320 est organisée aujourd'hui en cinq générations principales qui correspondent chacune à une classe de performances et d'applications.

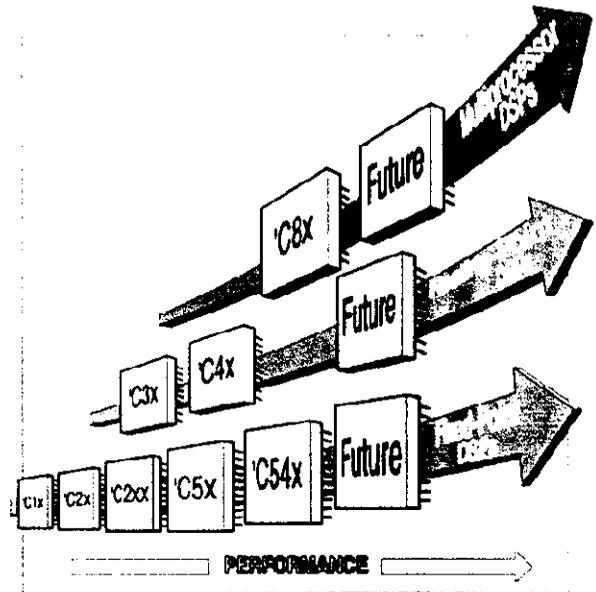
Les processeurs de traitement du signal ont vu leur utilisation se développer considérablement depuis 1985 et, ce, notamment avec l'évolution des télécommunications.

Dans cette partie, nous donnerons quelques évolutions des DSP (**Figure II.2**).

Nous parcourrons l'histoire des DSP de la première à la cinquième génération [2].

La famille DSP TMS320 est organisée aujourd'hui en cinq générations principales qui correspondent à une classe d'application et de performances qui sont :

- TMS320C1X,
- TMS320C2X,
- TMS320C3X,
- TMS320C4X,
- TMS320C5X.



**Figure II.2** Evolution des DSP, d'après [2].

#### I.4.1. Les DSP de la première génération (TMS320C1X)

Les DSP de la première génération sont semblables au TMS320C10.

- Tout d'abord la taille RAM (mémoire donnée) varie entre 144 et 256 mots.
- La taille de la ROM varie de 1.5K à 8K mots mais elle peut être remplacée par une EPROM de 4 ou 64 K mots.
- Certain de ces DSP peuvent contenir une ou deux entrées série au détriment d'une ou deux broches du port parallèle.
- La période de l'horloge peut varier de 280 à 114 nanosecondes.

#### I.4.2. Les DSP de la deuxième génération (TMS320C2X)

Dans cette deuxième génération de DSP, l'architecture interne a été nettement améliorée.

- Le bus de donnée a été séparé en deux. La première partie manipule les données en lecture et le second en écriture. Il dispose de 4 bus au total.

- Il traite seulement les nombres en virgule fixe.
- Le code source de cette deuxième génération est compatible avec ceux de la première génération.
- La durée d'un cycle d'horloge a elle aussi été diminuée, elle se situe entre 25 et 50 nanosecondes soit 20 à 40 millions d'opérations par seconde.
- La mémoire a plus que doublée, de plus le programme est stocké dans une ROM ou une mémoire FLASH.
- La pile comporte désormais 8 niveaux.
- Gère les interruptions.
- Les instructions sont encore plus rapides à exécuter. Il est possible par exemple de répéter plusieurs fois une instruction.
- Contient un port série synchrone ou asynchrone.

#### **I.4.3. Les DSP de la troisième génération (TMS330C3X)**

- Cette troisième génération de DSP fonctionne sur 32 bits et traite les données au format virgule flottante.
- l'utilisateur peut effectuer 11 opérations en même temps.
- double accès mémoire.
- Il dispose d'un ou deux canaux DMA (accès direct à la mémoire).
- L'espace mémoire intégré n'a pas extrêmement augmentée mais a été remanié.
- Ils disposent maintenant d'une séquence de lancement et d'une mémoire cache.
- Il a été conçu pour exécuter des algorithmes complexes.

#### **I.4.4. Les DSP de la quatrième génération (TMS340C4X)**

- Une RAM de 2kmots.
- 128 mots réservés pour le programme.
- 4Gmots accessibles sur 2 bus externes.
- 30 millions d'opérations par seconde.
- 488 Mbits par seconde de capacité E/S.
- 6 ou 12 canaux DMA.
- 6 canaux de connections multiprocesseurs.

### I.4.5. Les autres DSP de Texas Instruments

Il existe deux autres séries de DSP conçus par Texas Instruments :

Les TMS320C5X sont parmi les tous derniers DSP de chez Texas Instruments. Ils traitent des données au format virgule fixe. Leur conception les rend encore plus puissant, d'une part par leur mémoire interne et d'autre part par leur jeu d'instructions toujours plus puissant. Leurs bus DMA sont aussi une raison à leur puissance.

Les TMS320C8X ne sont ni des DSP virgule fixe, ni des DSP virgule flottante. Ce sont des DSP multiprocesseurs.

Les familles les plus récentes des DSPs de Texas Instruments sont :

- TMS320C54X, DSP format fixe ;
- TMS320C20X, DSP format fixe ;
- TMS320C24X, DSP format fixe ;
- TMS320C62X, DSP format fixe à architecture VLIW (Very long instruction word);
- TMS320C54X, DSP format flottant à architecture VLIW;

Ces nouvelles familles sont regroupées en 3 classes appelées plates-formes. Ces trois classes sont appelées : TMS320C2000, TMS320C5000 et TMS320C6000.

Le tableau suivant résume les principales caractéristiques de ces trois classes :

Application		Type de DSP	Caractéristiques
<b>TMS320C6000 DSP hautes performances</b>			
C62x	Applications exigeantes en vitesse : stations de base des réseaux de communications mobiles, équipement de radiodiffusion, réseaux informatiques	16 bits virgule fixe architecture VLIW	1200-2400 MIPS
C67x	Applications exigeantes en précision, dynamique et vitesse : Antennes adaptatives des stations de base, imagerie médicale, reconnaissance de parole, graphisme ...	32 bits virgule flottante architecture VLIW	600MFLOPS-1GFLOPS
<b>TMS320C5000 DSP optimisés en consommation</b>			
C54x	Applications de télécommunications exigeantes en coût, consommation, vitesse : terminaux mobile, voix sur IP, alphaspages ...	16 bits virgule fixe	0,54 mW/MIPS 30-200 MIPS
<b>TMS320C2000 DSP optimisés pour les applications de contrôle</b>			
C20x	Applications de grands volume en téléphonie, électronique grand public, appareils photos numériques ou contrôleurs de disques durs ...	16 bits virgule fixe	PLL, UART, timers, mémoire flash intégrée 20-40MIPS
C24x	Applications de contrôle moteur, automatisation, robotique, contrôle d'appareils électroménagers. ... Bon compromis prix/performance	16 bits virgule fixe	Port série SCI, SPI et CAN, Convertisseur Analogique /numérique 20MIPS

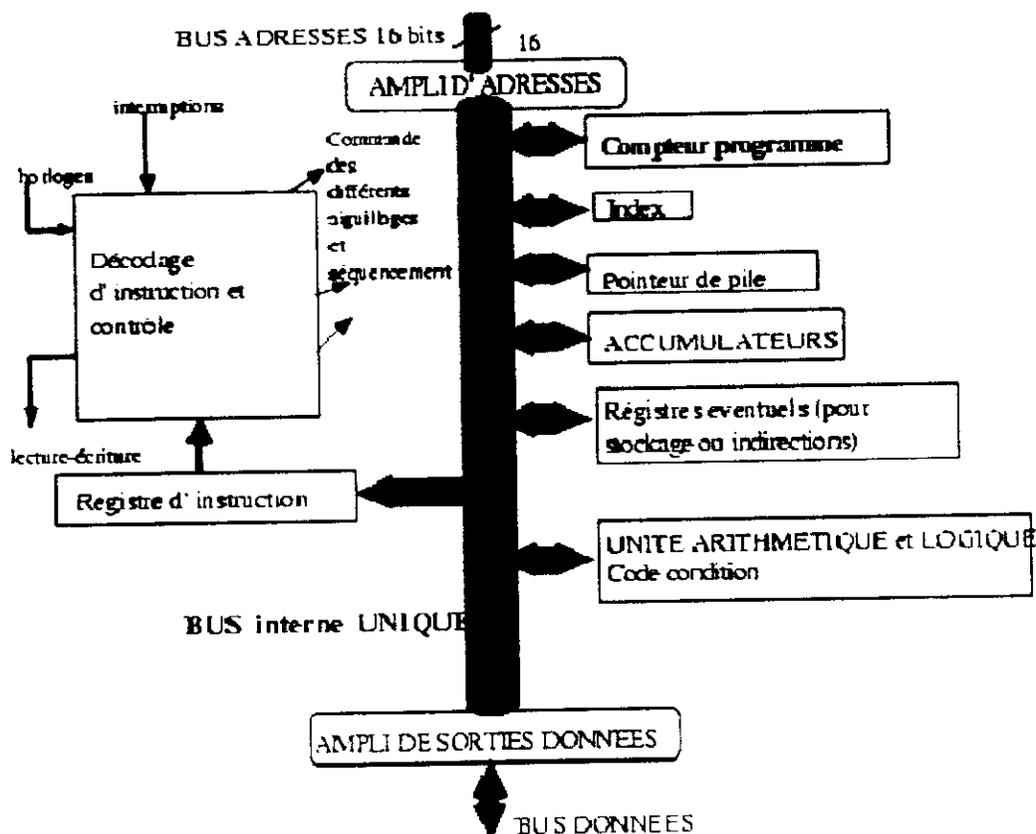
**Tab.1** : Principales caractéristiques des 3 familles de DSP TI

## 1.5 Les différences entre les DSP et les microprocesseurs

La différence fondamentale se situe sur le simple fait qu'un microprocesseur n'est pas destiné à une application spécifique alors que le DSP est destiné au traitement du signal ce qui lui permet d'avoir une architecture optimisée pour ces types de traitements. Nous allons ici vous introduire le fonctionnement d'un microprocesseur standard et développer les parties de celui-ci que nous pourrions optimiser dans le cadre du traitement du signal.

### 1.5.1 Microprocesseur standard

#### 1.5.1.1 Architecture interne



### 1.5.1.2 Déroulement d'une instruction (appliquée au 68000)

Les instructions sont placées dans des cases consécutives en mémoire puis sont exécutées en séquence. Une exécution se déroule en deux phases :

1. La recherche de l'instruction (fetch);
2. L'exécution (execute);

Un microprocesseur standard a besoin au minimum de 3 à 4 périodes afin d'exécuter une instruction. Le tableau ci-dessous décrit les différentes phases nécessaires à l'exécution d'une instruction simple.

Phases	Bus d'adresse	Bus de donnée	Cycle
Recherche et chargement dans le registre d'instruction. Décodage des codes opérations.	adresse du code opération en mémoire programme.	code instruction.	fetch
Décodage et calcul de l'adresse des opérandes.	Adresses suivantes en mémoire programme.	adresse de l'opérande	execute
Chargement dans le registre des opérandes.	adresse de l'opérande en mémoire donnée.	donnée	execute
Positionne l'ALU Exécute l'opération. Sauvegarde des résultats			execute

### 1.5.1.3 Réduction du nombre de périodes

Afin de réduire le nombre de phases et donc le nombre de périodes d'horloge nous pouvons effectuer plusieurs optimisations :

- Réduire à une adresse l'instruction complète;
- Envisager un processeur capable de superposer plusieurs phases, c'est la structure "pipe line" décrite ci-dessous.

Cette structure n'est envisageable que si l'on peut lire en même temps la mémoire donnée et la mémoire programme.

Nous devons donc avoir une structure avec deux bus séparés (la structure Harvard) ainsi que deux zones mémoires (mémoire programme et mémoire données) entièrement distinctes.

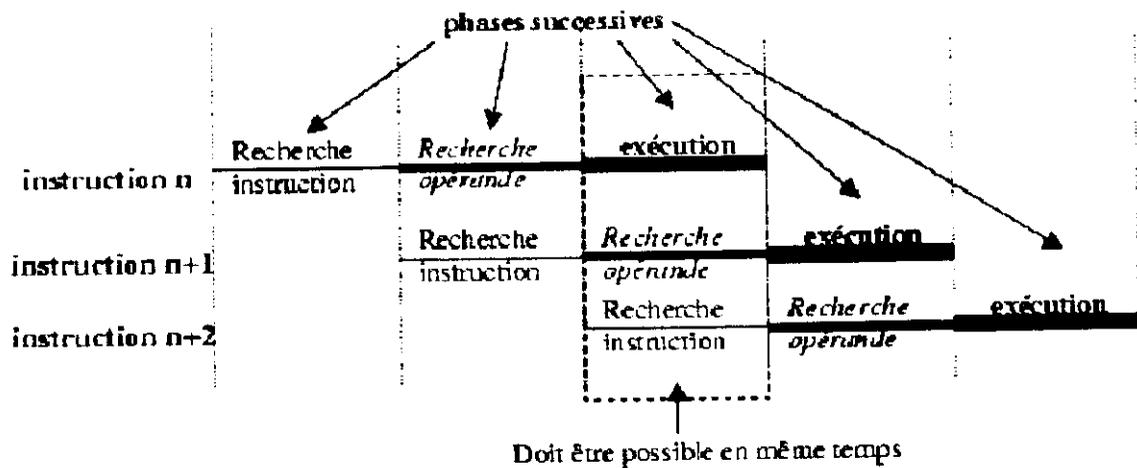


Figure I.3 :structure Pipe Line.

## I.6 Architecture externe

### I.6.1 Schéma général externe

Nous allons définir ici le schéma général d'utilisation d'un DSP qui permet de développer la plupart des applications simples en traitement du signal.

Les circuits exploitant les processeurs de traitement du signal ont une architecture particulière décomposable en deux parties :

- Une chaîne d'acquisition.
- Une unité de traitement du signal (DSP).

### I.6.2 La chaîne d'acquisition

Le signal en entrée du circuit peut être :

- un signal électrique ;
- une grandeur physique par le biais d'un capteur ;
- un signal numérique ;

Puis en fonction du signal d'entrée plusieurs éléments peuvent constituer la chaîne d'acquisition.

**Dans le cas d'un signal électrique ou d'une grandeur physique :** Le signal rencontrera en premier lieu un amplificateur d'adaptation qui comme son nom l'indique permet d'adapter le signal en décalage et en gain afin qu'il soit compatible

avec les autres éléments du circuit. Puis le signal arrive à l'échantillonneur-bloqueur qui permet de discrétiser le signal en prélevant des portions à des intervalles de temps réguliers avant de les envoyer au convertisseur analogique-numérique. Après un traitement en "temps réel" le DSP renvoie plusieurs signaux en direction soit d'un ordinateur (par le biais de la sortie numérique) soit vers un convertisseur numérique-analogique [8].

**Dans le cas d'un signal numérique :** Les DSPs présentant une entrée numérique sont aptes à les traiter directement.

### 1.6.3 Schéma

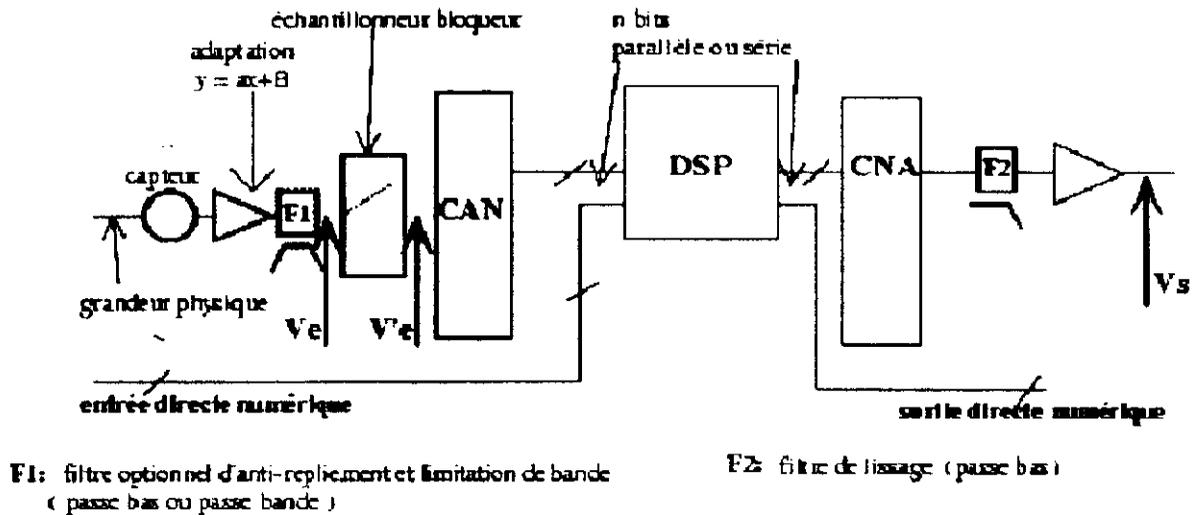


Figure 1.4 :Schéma général d'utilisation d'un DSP.

## Conclusion

Cette étude a mis en avant certaines notions de bases des DSP. Toutefois, comme précisé dans l'introduction, les DSP ne sont qu'une partie du traitement numérique du signal. Plus exactement ils en sont un des supports clés pour la réalisation pratique. De ce fait, programmer un DSP fait appel à un large champ de connaissances techniques (électronique, informatique, traitement du signal). Un traitement numérique du signal avec un DSP trouve des applications dans des domaines de plus en plus variés (exemples : asservissement des suspensions d'une automobile, réalisation d'un modem «logiciel», compression d'image vidéo ...etc.). Le principal moteur actuel de développement et d'évolution des DSP est l'immense marché des téléphones cellulaires (les « sans-fils »). Le DSP y réalise principalement la

compression / décompression de la voie humaine, permettant de la transmettre avec un faible débit numérique.

L'utilisation des DSP trouve ses limites lorsqu'il s'agit de traiter en temps réel des signaux de fréquences très élevés, ou lorsqu'il s'agit de réaliser des fonctions trop simples pour justifier la complexité et le coût de l'électronique nécessaire à sa mise en œuvre.

Sous quelque forme que ce soit, l'utilisation des DSP, et donc du traitement numérique du signal, constitue un domaine qu'il devient de plus en plus indispensable de connaître, et si possible de maîtriser.

# Chapitre:II

---

Caractéristiques techniques de

dSPACE DS1104

## II. Introduction :

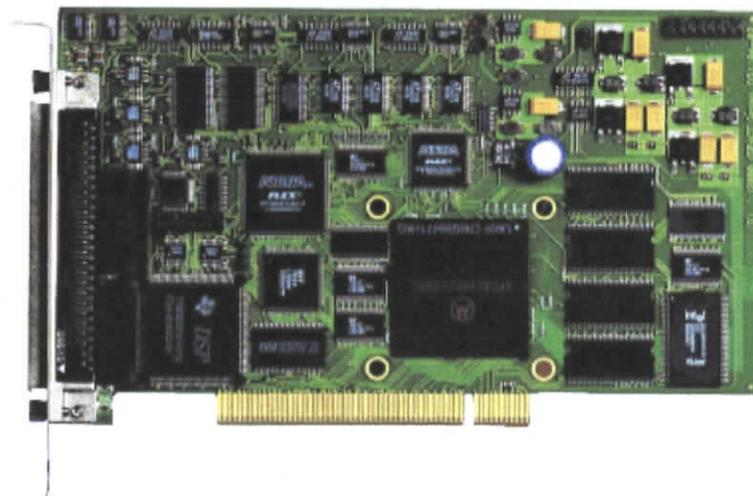
Dans ce présent chapitre, nous allons nous intéresser aux différentes architectures qui caractérisent l'ensemble des processeurs intégrés dans notre carte DS1104 et nous allons aussi faire une présentation du matériel dont elle dispose.

### II.1. Présentation de la carte d'acquisition DS1104 de dSPACE

La carte qu'on a utilisée est la carte **DS1104**, elle est assemblée et conçue par la société **dSPACE**.

#### II.1.1. Vue d'ensemble de la carte :

Cette carte possède des processeurs rapides et offre une sélection complète d'interface d'E/S typiques. Elle s'intègre directement dans un PC via le bus PCI.



**Figure II.1** : la carte DS1104.

#### II.1.2 Composition de la carte DS1104 :

La carte DS1104 est une carte très performante comportant deux processeurs et disposant de nombreuses entrées / sorties analogiques et numériques. Ainsi que des encodeurs incrémentaux pour la commande temps réel des moteurs.

Les différentes unités dont notre carte dispose sont :

**Processeurs** : La carte DS1104 est un système basé sur le processeur Maître (Master) Motorola MPC8240 avec un noyau 603e PPC (250 MHz clock), qui représente l'unité principale de traitement. Elle est aussi basée sur le DSP TMS320F240 (2<sup>ème</sup> génération) qui constitue l'unité principale de traitement. Elle est ainsi idéalement adaptée aux algorithmes numériques les plus complexes. Elle contient une mémoire globale (SDRAM) de 32 MB suffisamment rapide (ne nécessitant pas de temps d'attente : « zero wait state ») et une mémoire flash de 8 MB pour les applications en temps réel. Autour de cette carte DS1104, nous retrouvons tout ce qui est nécessaire au développement et à la mise en œuvre de cette carte, qui comprend un ensemble de périphériques (intégrés sur la carte) les plus utilisées en contrôle numérique ou en traitement numérique du signal des systèmes tels que les convertisseurs Analogique / Digital (ADC) et Digital / Analogique (DAC), signaux M.I., interfaces séries, décodeur incrémental, entrées / sorties numériques...etc.

Chaque processeur possède son compilateur C approprié (un pour PPC Motorola et un pour Slave DSP de Texas-Instruments).

**Unités d'Entrée / Sortie** : Un ensemble de périphériques à bord ; fréquemment utilisé dans des systèmes de commande numérique a été ajouté au PPC. Aussi bien que les convertisseurs A/D et D/A ,les ports numériques d'E/S , l' interface série, PPC a aussi la commande des encodeurs incrémentaux, qui permettent le développement des contrôleurs avancés pour des robots.

**Le sous ensemble DSP** : est un processeur esclave (Slave) basé sur le TMS320F240 de Texas Instruments. Il est particulièrement conçu pour la commande des moteurs électriques. Parmi autres, Les possibilités d'Entrée/Sortie, le DSP fournit la génération triphasée de PWM faisant un sous-ensemble utile pour le contrôle et le guidage d'applications.

Le PPC a accès au DSP et aux sous-ensembles des CAN. Parlons en termes de communication d'inter processeur, le PPC est le maître, tandis que le DSP est l'esclave.

En tout notre carte se compose de :

- Un Microprocesseur maître MPC8240 (Motorola) et un DSP esclave TMS320F240 (Texas-Instruments).
- Une mémoire SDRAM 32 MO, une mémoire flash 8 MO et un port-dual RAM (4 K mots).
- Une interface série (RS232 / RS485 / RS422).
- Unité d'E/S numérique (20 bits) et Une unité d'E/S numérique (14 bits).
- Deux interfaces d'encodeurs incrémentaux.
- 5 convertisseurs analogiques-numériques et 8 convertisseurs numériques-analogiques.

- 6 timers.
- Générateur PWM monophasé et triphasé.
- Quatre unités de captures.
- Un bus PCI, permet la connexion entre la carte et le connecteur d'E/S (P1) [8].

La figure suivante donne une vue d'ensemble des unités fonctionnelles de la carte DS1104.

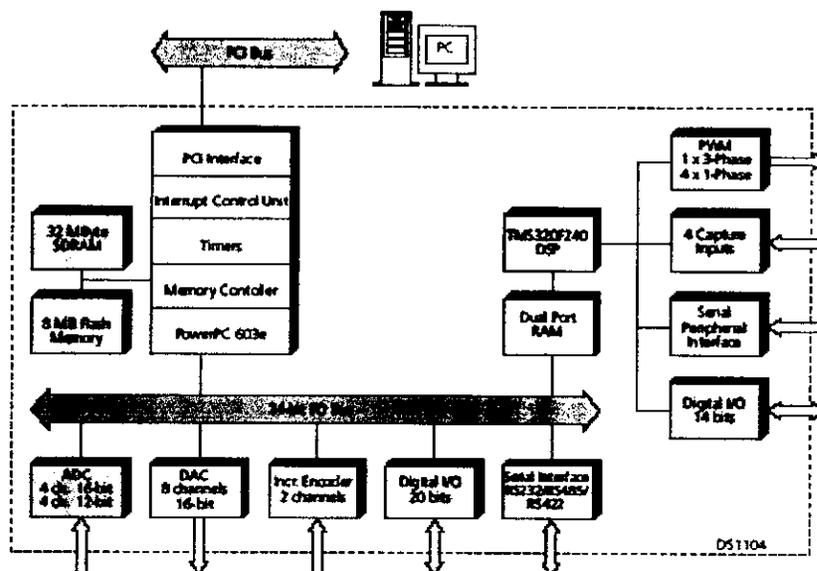


Figure II.2 : Vue d'ensemble de la carte DS1104 et ses différents composants [6].

## II.2 Le processeur maître MPC8240 :

Le processeur MPC8240 est construit par la firme MOTOROLA.

Le MPC8240 combine un noyau PowerPC 603e avec un bus PCI, il intègre un contrôleur mémoire de haute performance qui supporte les mémoires DRAM et ROM. Il est choisit dans cette carte pour sa faible consommation, sa taille réduite et sont faible coût. Il a deux horloges PLL différentes pour chaque un des deux blocks.

Ce processeur peut être utilisé dans plusieurs applications telle que :

- La télécommunication.
- La commande des appareils de réseaux.
- La commande des systèmes d'impressions.

Mais l'une de ces applications qui nous intéresse est la gestion et le contrôle du DSP esclave.

Le MPC8240 est décomposé en deux blocks principaux :

- Le noyau PowerPC 603e.
- Périphérique logique.

### II.2.1 Paramètres généraux :

La liste suivante fournit un sommaire des paramètres généraux du MPC8240 :

- Technologie : 0.29- $\mu$ m CMOS, métal de cinq couches.
- Taille : 73 mm<sup>2</sup> de matrice.
- Nombre de transistors : 3.1 millions.
- Conception de logique Plein-statique.
- 352 broches.
- Alimentation d'énergie du noyau : 2.5V  $\pm$  5% V.
- Alimentation d'énergie d'E/S : de 3 à 3.6 V.
- La fréquence du CPU est de 250 MHz.

### II.2.2 Dispositifs principaux du MPC8240 :

Les dispositifs principaux du MPC8240 sont les suivants :

#### II.2.2.a Périphériques logiques :

Les périphériques du processeur maître les plus importants sont :

- L'interface PCI : C'est une interface parallèle de 32 bits actionnant jusqu'à 33 MHz, elle a une tolérance de 5V  $\pm$  5%.
- Contrôleur d'interruption programmable (PIC) : Il permet de sélectionner les interruptions de masquages (cinq interruptions matériel « IRQs » ou 16 interruptions séquentielles et Quatre Timers programmables).
- Contrôleur mémoire.
- Contrôleur DMA intégré à deux voies (écriture en ROM / Port X non supporté).
- Contrôleur I<sup>2</sup>C : C'est un bus spéciale qui permet d'établir une liaison série synchrone entre 2 ou plusieurs composants se trouvant sur la même carte. Il permet aussi de transmettre l'information du processeur maître au DSP avec un protocole défini.
- Contrôleur de message I<sup>2</sup>O.
- Bus de haut largeur de bande (bus 64 bits de données).
- Bus d'adresses de 32 bits.

**N.B :** La communication entre ces deux blocks est possible via un bus périphérique logique de 32 bits.

La figure suivante nous montre son bloc diagramme :

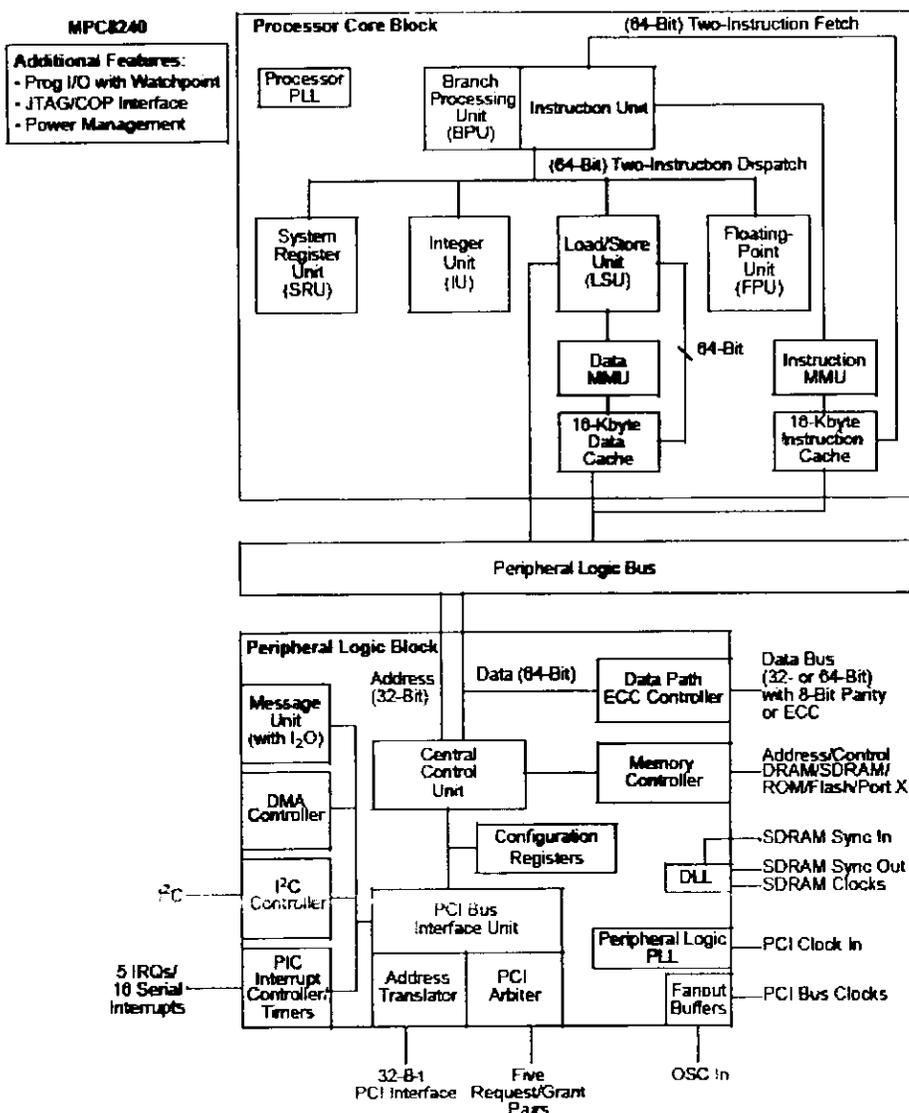


Figure II.3 : diagramme bloc du processeur maître MPC8240[17].

### II.2.2.b Le noyau PPC 603e :

Le powerPC 603e est le second membre de la famille créée par Apple-IBM-Motorola après le PowerPC601e. Il s’aligne sur la tension d’alimentation normalisée de 3.3 V. Puis, il dispose de plusieurs modes d’économie d’énergie répondant aux nouvelles exigences du programme américain « Energy Star ». Sa consommation peut ainsi être réduite au niveau du watt, ce qui en fait un précieux auxiliaire pour les portables.

Il est composé de plusieurs unités [3]:

-Unité de traitement de branchement **BPU**: elle possède un compteur ordinaire (CO), qui contient l'adresse de l'instruction à chercher et exécuter, il cherche deux instructions de 32 bit chacune.

-Unité d'instruction : elle reçoit l'instruction qui est transmise par la mémoire et elle détermine l'adresse de la prochaine instruction à partir des informations de l'unité de recherche séquentielle et de l'unité de branchement.

-Unité de gestion de la mémoire **MMU** : elle travaille sur une adresse virtuelle de 52 bits et sur une adresse réelle de 32 bits. Huit blocs de registres de translation d'adresses apparaissent. Les registres de translation d'adresses, ou **TLB** (buffer de translation d'adresse) sont sur 64 entrées deux voies pour les instructions et les données. Il s'agit des :

- Buffer de translation de données (**DTLB**).
- Buffer de translation d'instruction (**ITLB**).

La translation d'adresse agit sur des pages de 4 Ko, des blocs de longueur variable ou des segments de 256 Mo.

-Unité de gestion de la mémoire instruction **MMU**.

-Unité de gestion de la mémoire donnée **MMU**.

-Unité de calcul sur entier **IU**: elle effectue des opérations arithmétiques et logiques sur des nombres entiers.

-Unité de calcul en virgule flottante **FPU**: elle effectue des opérations arithmétiques et logiques sur des nombres réels.

-Unité de registre système **SRU**.

-Unité de chargement / enregistrement **LSU**.

-Les TLB et les caches appliquent l'algorithme de la plus ancienne donnée utilisée, ou **LRU** qui s'occupe de la gestion des remplacements.

-Une mémoire cache de donnée 16 Koctet.

-Une mémoire cache d'instruction 16 Koctet.

-La PLL du CPU : elle génère un signal d'horloge de 250 MHz.

La figure suivante représente le synoptique simplifié du processeur MPC603. C'est un système 32 bits procurant :

- Des adresses effectives (logiques) sur 32 bits ou 64 bits.
- Des données entières sur 8 bits, 16 bits ou 64 bits.
- Des données en virgule flottante sur 32 bits ou 64 bits.

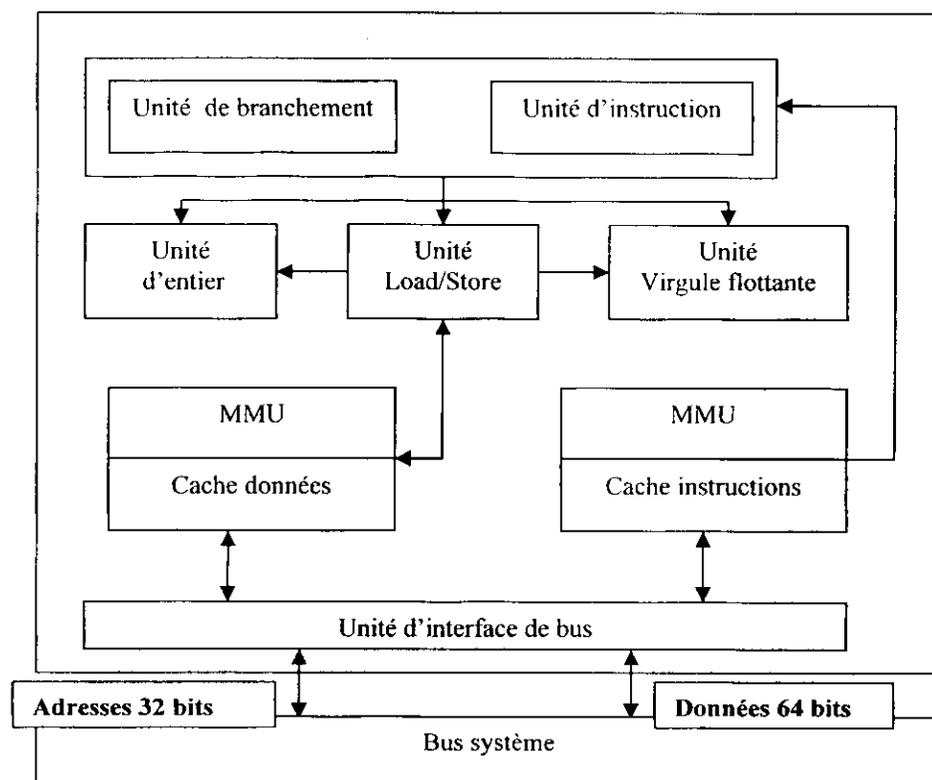


Figure II.4 : Synoptique simplifié du processeur PPC 603e.

### II.3 DSP esclave :

#### II.3.1 description :

Le DSP esclave est le TMS320F240 qui appartient à la firme TEXAS-INSTRUMENTS. C'est un membre de la famille des DSP de la génération TMS320C2XX ; 16 bits à virgule fixe. Il se caractérise par un noyau CPU de faible coût de conception, une performance de traitement, et une structure PIPELINE.

Ce contrôleur combine une architecture Harvard amélioré qui se caractérise par deux mémoires distinctes (mémoire données et mémoire programmes) accessibles par deux bus qui sont :

- le bus de données.
- le bus de programmes.

Cela permet la simultanéité du décodage d'instructions et de l'exécution. Il faut noter qu'on peut avoir un transfert de données entre les deux bus.

Cette famille est utilisée essentiellement pour la commande des moteurs.

**II.3.2 Brochage :**

Notre DSP contient 132 broches (analogue au TMS320C5X), comme le montre la figure suivante :

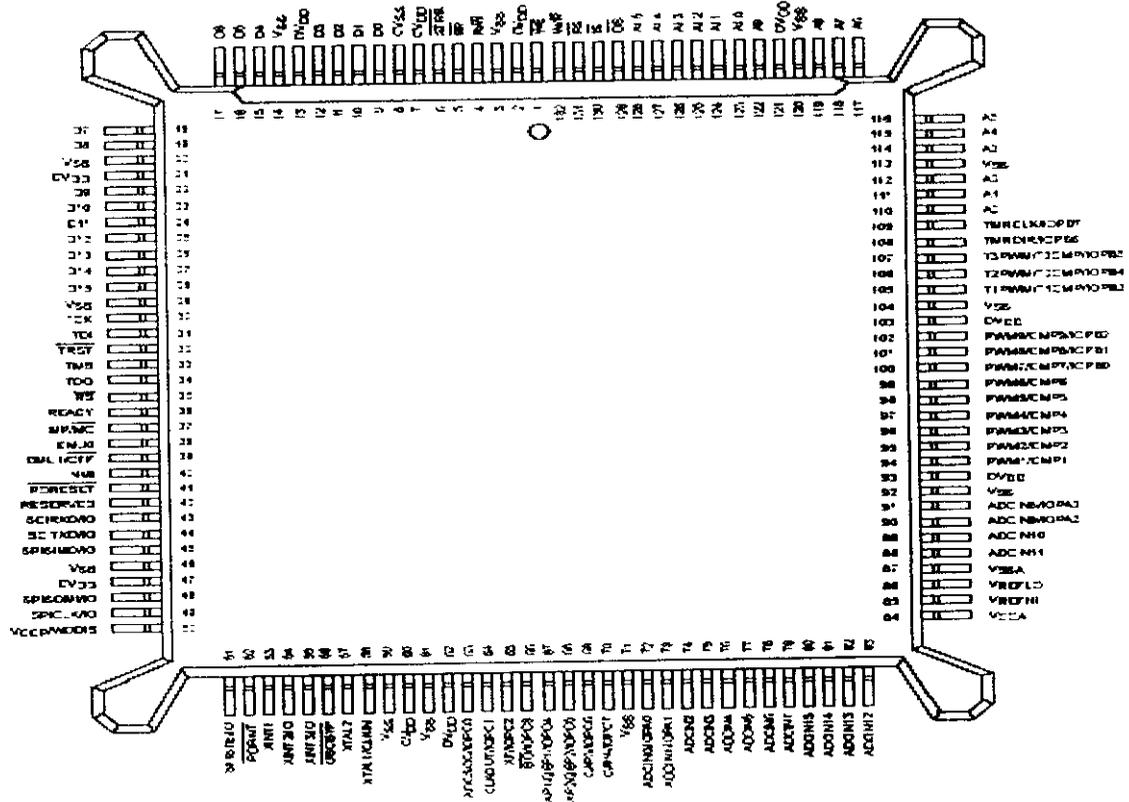


Figure II.5 : Brochage du TMS320F240 [9].

**II.3.3 Description des principales broches de DS1104 :**

TERMINAL		TYPE	DESCRIPTION
NOM	N°		
A0	110		Bus d'adresses
A1	111		
A2	112		

A3	114	O/Z.	C'est un bus de 16 bits de sortie, il permet d'adresser les mémoires donnée/programme externe ou les dispositif d'E/S .Il est placé au niveau haute impédance lorsque la broche $\overline{EMU1/OFF}$ est mis au niveau bas.		
A4	115				
A5	116				
A6	117				
A7	118				
A8	119				
A9	122				
A10	123				
A11	124				
A12	125				
A13	126				
A14	127				
A15	128				
D0	9			I/O/Z.	<p style="text-align: center;"><b>Bus de données</b></p> <p>Il a trois état : bidirectionnelle pour le transfert des donnée entre le DSP et la mémoire de donnée /programme externe, et les dispositifs d'E/S.</p> <p>Ce bus peut être placé en état haute impédance lorsque :</p> <ul style="list-style-type: none"> <li>- <math>\overline{RS}</math> est affirmé.</li> <li>- <math>\overline{EMU1/OFF}</math> est activé au niveau bas</li> <li>- Mode de mise hors tension du circuit.</li> </ul>
D1	10				
D2	11				
D3	12				
D4	15				
D5	16				
D6	17				
D7	18				
D8	19				
D9	22				
D10	23				
D11	24				
D12	25				
D13	26				
D14	27				
D15	28				
$\overline{DS}$	129	O/Z.	La sélection du type d'accès via le bus adresses et données se fait par ces signaux (donnée, programme et I/O strobe). Ils sont placé en haute impédance pendant le reset, mode de		
$\overline{PS}$	131				
$\overline{IS}$	130				

			mise de hors tension du circuit, et $EMU1/\overline{OFF}$ est activé au niveau bas.
READY	36	I	En 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 teste le ready. il aussi utilisé avec le $\overline{BR}$ .
$\overline{BR}$	5	O/Z	Sortie qui indique que le DSP demande l'accès à la mémoire donnée externe, comme il peut être utilisé pour étendre l'adressage de la mémoire donnée jusqu'à 32 K mots. il est mit à l'état haute impédance pendant le reset, mode de mise de hors tension du circuit, et $EMU1/\overline{OFF}$ est activé au niveau bas.
$R/\overline{W}$	4	O/Z	Elle est en sortie, elle indique le sens de transfert des données, mais il faut qu'il soit confirmé par la broche $\overline{WE}$ (si $\overline{WE}$ est au niveau bas, cela indique que l'opération d'écriture est permise, sinon c'est l'opération de lecture qui est permise).
$W/\overline{R}$	132	O/Z	Il diffère du premier car il permet de valider l'opération d'écriture et de lecture directement.
Vccp/WDDIS	50	I	En entrée, en peut y accédé à cette broche et la programmer par un interrupteur qui se trouve dans la carte DS1104, et qui peut être mis soit à Vcc (par défaut) et dans ce cas l'opération d'écriture et d'effacement de la mémoire interne est validé, ou en GND (mise à la masse) et dans ce cas l'opération est inhibée.
ADCIN0	72	I	Entrée analogique pour le premier ADC
ADCIN1	73		
.....			
ADCIN7	79		
ADCIN8	91		

.....		I	Entrée analogique pour le second ADC
ADCIN15	80		
$\overline{STRB}$	6	O/Z	Ce signal indique un accès externe
PWM1/CMP1	94	O/Z	Sortie PWM ou de comparaison, ces signaux sont générés par le gestionnaire d'évènement q'on va voir après.
PWM2/CMP2	95		
PWM3/CMP3	96		
PWM4/CMP4	97		
PWM5/CMP5	98		
PWM6/CMP6	99		
PWM7/CMP7/IOPB0	100	I/O/Z	Entrée sortie numérique, il est généré par le même périphérique.
PWM8/CMP8/IOPB1	101	I/O/Z	Entrée sortie numérique, il est généré par le même périphérique.
PWM9/CMP9/IOPB2	102	I/O/Z	Entrée sortie numérique, il est généré par le même périphérique.
T1PWM/T1CMP/IOPB3 105		I/O/Z	Le timer 1 du EV qui génère ce signale
T2PWM/T2CMP/IOPB4 106		I/O/Z	Le timer 2 du EV qui génère ce signale
T3PWM/T3CMP/IOPB5 107		I/O/Z	Le timer 3 du EV qui génère ce signale
$\overline{RS}$	35	I/O	$\overline{RS}$ en entrée cause l'arrêt de l'exécution d'instruction par le DSP, et place le PC, les registres, et les bits d'états à zéro. Si $\overline{RS}$ est au niveau haut : exécution du programme à l'adresse zéro.
MP/ $\overline{MC}$	37	I	C'est une broche de sélection du mode de fonctionnement du DSP : -si MP/ $\overline{MC}$ est au niveau bas, la mémoire programme interne est sélectionné -si MP/ $\overline{MC}$ est au niveau haut, la mémoire

			programme externe est sélectionné.
NMI	40	I	C'est l'interruption la plus prioritaire après le RESET, et va interrompre le processeur indépendamment de l'état du bit INTM du registre d'état.
$\overline{PORSET}$	41	I	C'est une interruption qui cause l'arrêt d'exécution des instructions par le F240, place le PC=0, et initialise les registres de contrôle de la PLL.
XINT1	53	I	Interruption utilisateur externe N°1 masquable.
XINT2/IO	54	I/O	Interruption utilisateur externe N°2 masquable.
XINT3/IO	55	I/O	Interruption utilisateur externe N°3 masquable.
$\overline{PDPINT}$	52	I	Interruption masquable ou non masquable.

### II.3.4 Architecture du TMS320F240 :

Notre DSP est se compose de trois unités fonctionnelles principales :

- noyau du DSP C2XX (CPU).
- mémoire interne.
- périphériques.

En plus de ça on a d'autres dispositifs :

- le dispositif d'interruption.
- la carte mémoire.
- le reset.
- le dispositif numérique d'E/S.
- l'horloge.

La figure suivante nous donne son diagramme bloc:

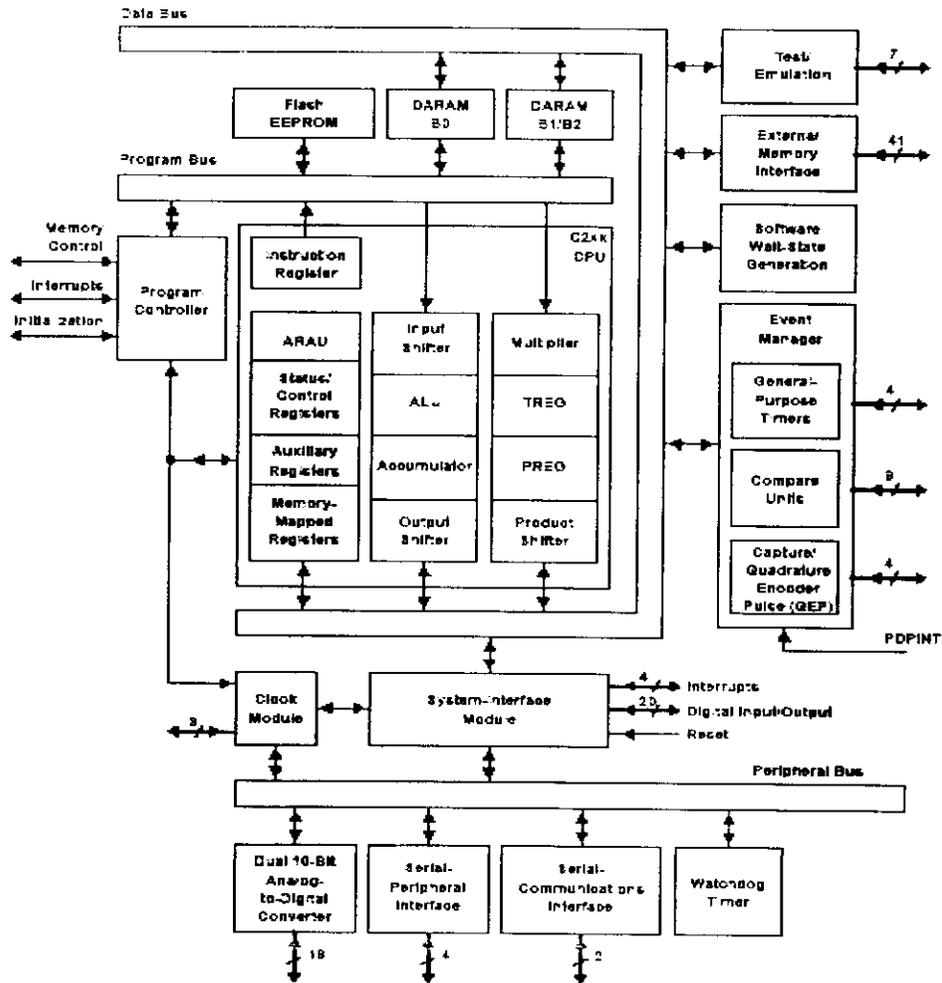


Figure II.6 : Le bloc diagramme fonctionnel du TMS320F240 [9].

### II.3.5 Cartographie mémoire :

Le F240 implémente trois espaces de mémoire différentes :

- mémoire donnée.
- mémoire programme.
- espace d'E/S.

Chacun a une capacité de 64 Kmots.

On peut mapper plusieurs composants :

- Registres.
- Périphériques.
- mémoires.

La figure suivante nous donne une idée générale :

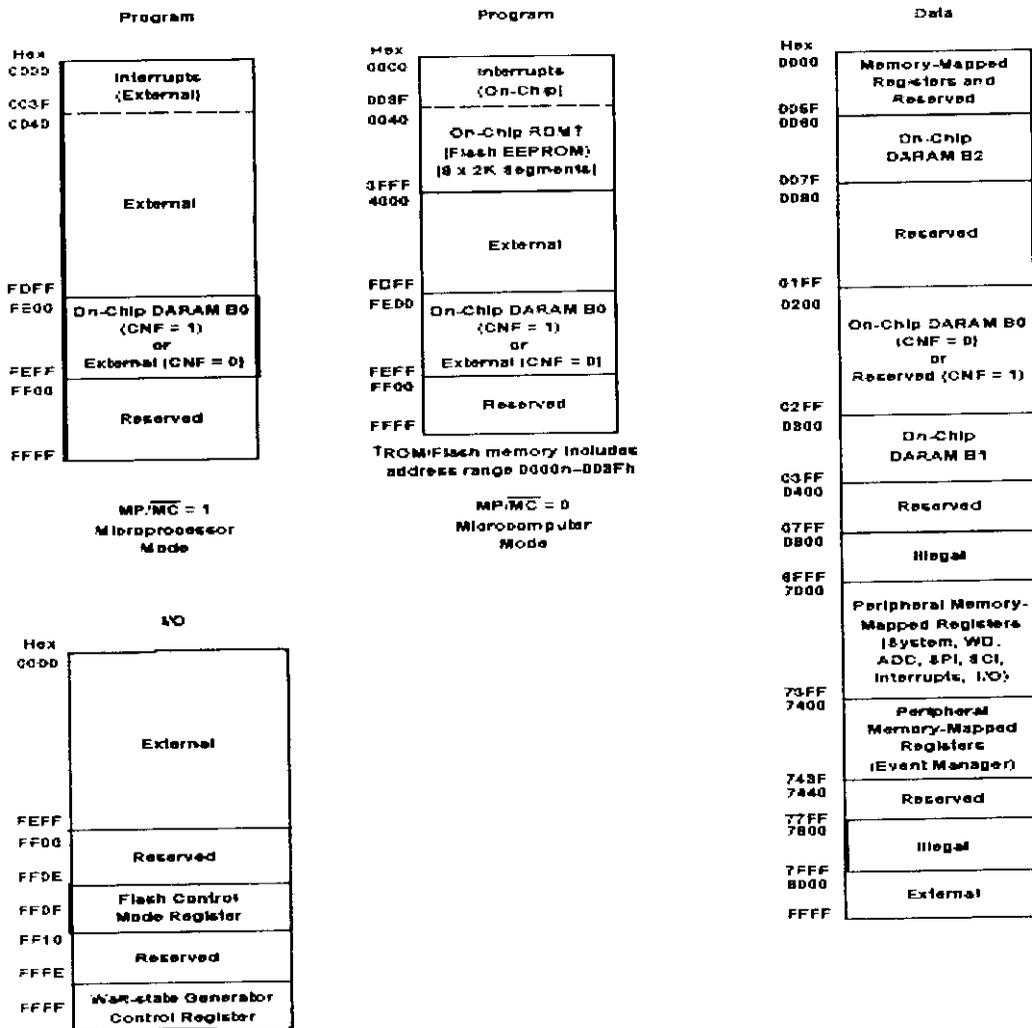


Figure II.7 : Cartographie mémoire [9].

### II.3.6 CPU (unité centrale de traitement) :

La CPU est composée d'une unité de calcul arithmétique et logique (CALU), d'un multiplieur, décaleur d'entrée ,2 décaleurs de sortie, et d'un accumulateur.

Comme la montre la figure suivante :

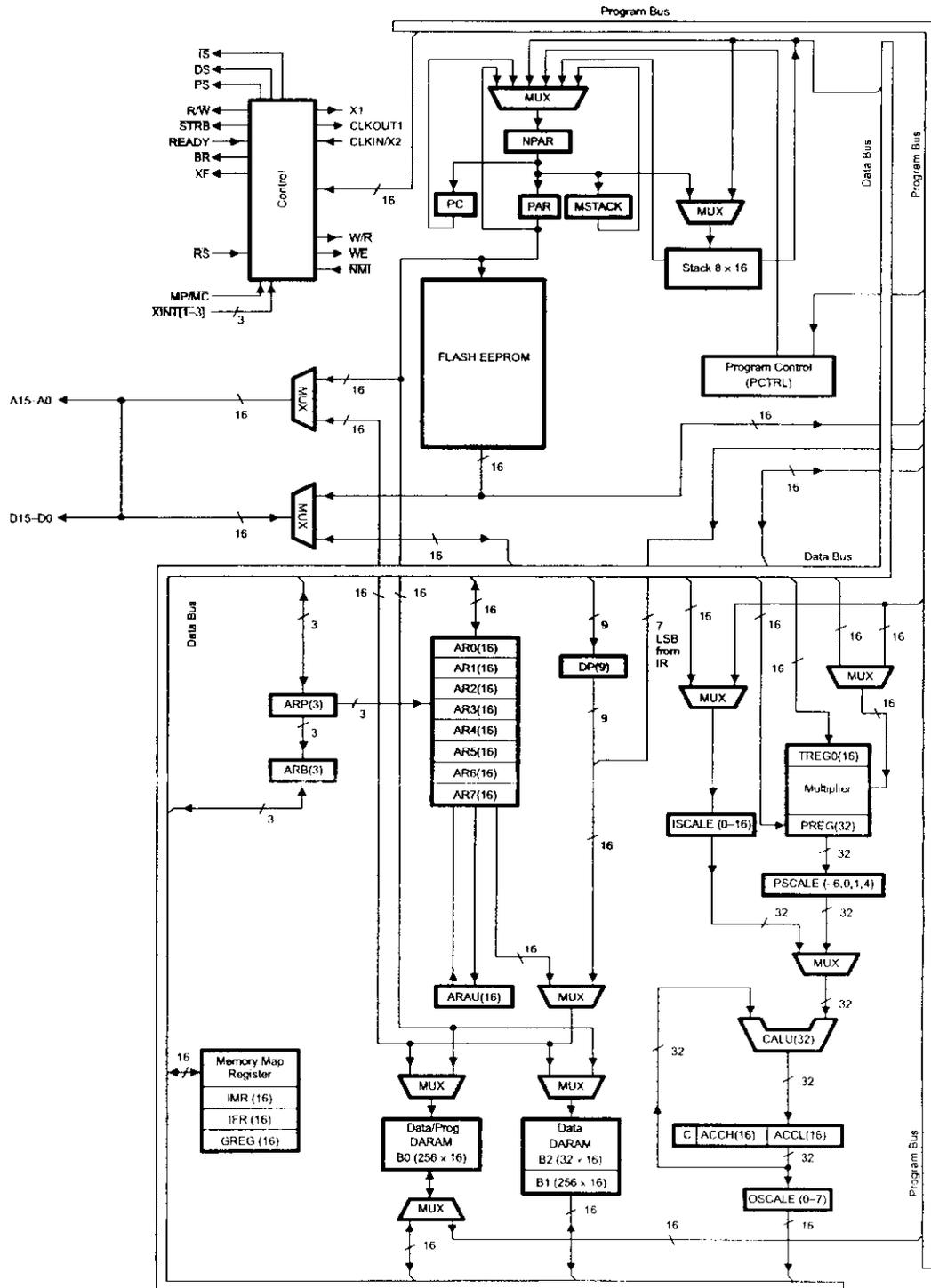


Figure II.8 : Bloc diagramme de la CPU.

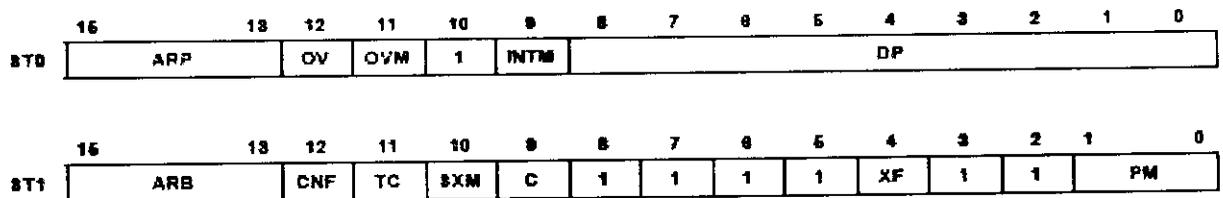
**II.3.6.1 décaleurs :**

il existe 2 décaleurs ,l'un est le décaleur d'entrée 16 bit connecté au bus de donnée (ISCALE),qui fait un décalage de 0 à 16 bit à gauche telle que les bit LSB sont mit à zéro avec extension de signe .

Alors que l'autre est le décaleur de sortie (OSCALE) 32 bits connectés à la CALU et qui fait un décalage de 0 à 7 bit à gauche.

**b-registres d'état :**

Le CPU contient 2 registres d'états ST0, et ST1, de 16 bit chacun comme le montre la figure suivante :



Ces deux registres d'états peuvent être enregistré en mémoire donnée.

Le tableau suivant nous définit chaque bit :

ZONE	FONCTION
ARB	Auxiliary register pointer buffer (3 bit) Lorsque ARP est chargé dans ST0, ARB va être chargé par l'ancien contenu de ARP.
ARP	Pointeur de registre auxiliaire (3 bits) ARP sélectionne le registre auxiliaire (AR) qui va être utilisé pour l'adressage indirect.
C	Bit carry, si le résultat de l'addition dans ACC génère un retenu, il est mis à 1, sinon il est remis à 0.
CNF	C'est un bit de contrôle de la configuration de la puce RAM. Si CNF est mis à 0, le block B0 est choisi comme mémoire donnée, sinon il est choisi comme mémoire programme.
DP	Pointeur de page de la mémoire donnée (9 bits) Les 9 bit de DP sont concatène avec les 7 bit LSB du bus programme pour formé une adresse mémoire de

	16 bits. DP peut être modifié par les instructions LST ou LDP.
INTM	Bit de mode d'interruption. Si INTM est mis à 0, les interruptions non masquables vont être valides par le MP. Si INTM est mis à 1, toutes les interruptions masquables vont être inhibées. INTM est placé et remis par les instructions : SETC INTM et CLRC INTM.
OV	Bit indicateur surcharge. le MP met à 1 le bit OV lorsqu'il y'a un débordement dans l'unité arithmétique et logique (CALU).
OVM	Un bit pour déterminer le mode de saturation. Ce mode peut être de type complément à 2
PM	2 bits qui indiquent le nombre de bits décalés et leur sens (droite ou gauche).
SXM	Bit de mode extension de signe. Si SXM est mis à 1 on travaille en mode où la donnée est étendue, sinon l'extension de signe est supprimée.
TC	Bit indicateur de test et de control.
XF	Bit qui donne l'état de la broche XF.

**Tab. 1 : contenu des différents registres**

### II.3.6.2 multiplieur :

Le TMS320F240 dispose d'un multiplieur 16\*16 bits donnant un résultat sur 32 bits signé ou non signé dans un seul cycle d'horloge.

Toutes les instructions de multiplication excepte MPYU, exécutent des opérations de multiplication signées.

Les deux opérandes de multiplication plus le résultat sont des nombres en complément à 2.

Les deux registres qui sont associés avec le multiplieur sont :

-le registre temporaire 16 bits (TREG), qui porte une des opérandes de multiplication, alors que l'autre opérande vient soit du bus de donnée ou du bus programme.

-le registre de produit 32 bits (PREG), qui porte le résultat de la multiplication.

Comme on a déjà vu les deux bits PM du registre d'état ST1 qui permettent de connaître le nombre de bits décalés par le PSCALE à la sortie du PREG, et leur sens de décalage comme le montre le tableau suivant :

PM	décalage
00	Pas de décalage
01	1 bit décale à gauche
10	4 bits décalent à gauche
11	6 bits décalent à droite

**Tab. 2 : sens de décalage.**

Le décaleur d'un seul bit est utilisé pour compenser le bit de signe supplémentaire gagné par la multiplication  $2 \times 16$  bit par l'instruction MPY.

-le décalage de 4 bit est utilisé avec l'instruction MPY et qui multiplie un opérande de  $16 \text{ bit} \times 13 \text{ bit}$  pour éliminer les 4 bits supplémentaires de signe gagnés.

-le décalage avec 6 bits à droite permet l'exécution de 128 opérations de multiplication\accumulation, et les instructions MAC et MACD permettent de réaliser ça en un cycle d'horloge.

### II.3.6.3 Unité Arithmétique et logique centrale CALU :

L'unité centrale arithmétique et logique (CALU) exécute des opérations arithmétiques et logiques en un cycle d'horloge (50 ns).

L'un des opérandes vient de l'accumulateur (32 bits) toujours, alors que l'autre opérande de 32 bits vient soit du PREG après avoir subi un décalage par le PSCALE ou décale par le ISCALE, provenant soit du bus de données ou du bus de programme.

Après que la CALU exécute l'opération arithmétique (Addition ou soustraction) ou logique (Booléenne), le résultat est mis dans l'accumulateur sur 32 bits plus le retenu.

### II.3.6.4 Accumulateur :

Le résultat de l'opération effectuée par la CALU est enregistré dans l'accumulateur qui contient 32 bits, il peut être coupé en 2 mots de 16 bits qui peuvent être placés dans la mémoire donnée.

### II.3.6.5 Registre auxiliaire (AR) et unité arithmétique du registre auxiliaire (ARAU) :

Dans le CPU existe 8 registres auxiliaires de 16 bits, chacun utilisé pour l'adressage indirect de la mémoire donnée.

Ils sont sélectionnés par les 3 bits de l'ARP.

L'ARAU est connecté à ces registres, elle peut auto indexer et incrémenter le registre auxiliaire courant.

En plus de ça on a :

- Des multiplexeurs.
- Un registre de carte mémoire.
- IMR : registre de 16 bits de masque des différentes interruptions.
- IFR (Interrupt flag register) : registre de 16 bits qui mémorise les interruptions actives.
- GREG (Global memory allocation register) : registre 16 bits qui définit la taille de l'espace mémoire global.

#### II.4 La mémoire interne :

Le TMS320F240 contient deux types de mémoire internes :

- DUAL-access RAM (DARAM).
- EEPROM instantannée.

- **DARAM :**

Il y'a 544 mots de 16 bits dans la DARAM du F240, le F240 et les dispositifs externes peuvent écrire et lire sur la DARAM dans un même cycle.

La DARAM est configurée en trois blocs :

- block 0 (B0) : il contient 256 mots, il peut être configuré en mémoire donnée, en utilisant l'instruction SETC CNF, ou en mémoire programme par l'instruction CLRC CNF.
- block 1 (B1) : il contient 256 mots, localisés dans l'espace mémoire donnée.
- block2 (B2) : il contient 32 mots localisés dans la mémoire donnée.

- **Flash EEPROM :**

La mémoire EEPROM contient 16 kMots de mémoire programme. Cette mémoire flash augmente la capacité du F240 pour les zones de prototypage, les zones de texte, et l'application à microplaquette unique.

L'écriture et l'effacement sur cette mémoire se font de façon électrique.

## II.5 Le module d'horloge :

Le TMS320F240 contient le module d'horloge basé PLL, ce module fournit tous les signaux d'horloge nécessaire pour son fonctionnement.

L'unique composant externe nécessaire pour ce module est le cristal ou l'oscillateur.

La PLL a deux modes de fonctionnement :

a- mode oscillateur :

Ce mode permet d'utiliser un cristal de référence (quartz) de 4, 6 ou 8 MHz pour fournir un temps de base pour le DSP.

Le signal de sortie CPUCLK est généré par le logicielle.

b- mode d'horloge :

L'horloge du dispositif est générée par une source d'horloge externe qui entre dans la broche XTAL1/CLKIN.

Le F240 exécute sur deux fréquences d'horloge :

-CPUCLK.

-SYSCLK.

On ajoute à cela une autre fréquence d'horloge gènère par le timer du chien de garde WDCLK.

Ce module inclus trois broches externes :

-XTAL1/CLKIN : source d'horloge/entrée du cristal .

-XTAL2 : sortie du cristal .

- $\overline{OSCBYP}$  : il permet de validé ou non l'oscillateur.

## II.6 Périphériques :

Les périphériques intégrés dans le DSP sont :

-interface de mémoire externe.

-Gestionnaire d'événement (EV)

-convertisseur analogique-numérique (ADC).

-Interface périphérique séquentielle (SPI)

-Interface de transmission séquentielle (SCI).

-timer Watchdog (WD).

**a- Interface de mémoire externe :**

Le TMS320F240 peut adresser jusqu'à 64 kmots de 16 bit de mémoire ou de registre externe. Cette interface parallèle permet de multiplexer le bus de donnée en un bus de donnée et un bus d'adresse, il permet aussi le transfert des données, des mémoires ou dispositifs externes au DSP.

**b- Gestionnaire d'événements :**

Il est utilisé pour les applications de commande du moteur. Il est composé de :

- Trois timers d'usage universels.
- Trois unités de comparaison complète.
- Trois unités de comparaison simple.
- Les circuits de modulation de largeur d'impulsion (PWM), qui incluent :
  - un circuit d'espace vecteur PWM
  - trois unités de génération de bande morte
  - des sorties logiques
- Quatre unités de capture
- Circuit d'impulsion de quadrature encodeur.
- Des interruptions EV.

Les broches de EV sont :

## 1. broche PWM :

Ils sont aux nombres de 12, mais groupé en trois groupes :

## -Groupe1 :

Trois signaux Généré par les trois TIMERS :

- T1PWM/T1CMP
- T2PWM/T2CMP
- T3PWM/T3CMP

## -Groupe2 :

Six généré par l'unité de comparaison complète

- PWM1/CMP1.
- .....
- PWM6/CMP6.

## Groupe 3 :

Trois généré par l'unité de comparaison simple

- PWM7/CMP7.
- PWM8/CMP8.

--PWM9/CMP9.

2. broche de capture :

--CAP1/QEP1 et CAP2/QEP2 : ces entres sont utilisés pour interfacer le circuit QEP.

--CAP3 et CAP4.

En ajoute a ça :

TMRCLK qui fournit une entrée d'horloge externe.

TMRDIR qui est utilisé pour donnée la direction du comptage des timers, si 0, est dans ce mode. Il faut noter que ces trois timers peuvent être programmés pour des opérations basé sur l'horloge CPU interne ou externe.

### **c- Convertisseur analogique-numerique :**

Ce module contient 2 ADC telle que chaque ADC à 8 canaux d'E/S analogique, et après la conversion on obtient 10 bits dans le registre de contrôle.

Le temps de conversion pour chaque ADC est de 6.1µs.

### **d- Interface périphérique séquentielle (SPI) :**

Le F240 contient 4 broches qui sont reliées à ce module :

- SPISOMI.
- SPISIMO.
- SPISTE.
- SPI-CLK.

Cette interface permet la communication rapide entre le TMS320F240 et d'autre processeurs (MPC8240 dans notre cas).

### **e- interface de transmission séquentielle (SCI) :**

Le SCI permet la communication numérique entre le CPU et d'autre périphérique asynchrone. Ils possède deux bus l'un pour la transmission et l'autre pour la réception, mais il faut noter qu'on peut avoir la communication full duplex.

Pour assurer l'intégrité des données, le SCI examine les données reçus pour assure la détection, la priorité, le dépassement ...etc.

Il faut note que le SCI contient 2 broches externes :

-SCITXD : Broche externe de transmission.

-SCIRXD : Broche externe de réception.

### **f- WD et RTI :**

La fonction de WD est de surveiller le fonctionnement du logiciel et du matériel.

La fonction de RTI est de fournir des interruptions à intervalles programmables.

## II.7 Interruptions et RESET :

Le F240 reconnaît trois type de sources d'interruption :

**1-RESET** : qui peut être lancé du matériel (broche  $\overline{RS}$ ), ou du logiciel (instruction), c'est l'interruption la plus prioritaire, vectorisé a l'adresse 0.

Cette interruption peut être causée par :

- activation de la broche  $\overline{RS}$ .
- activation de la broche  $\overline{PORESET}$ .
- le RESET du timer Watchdog.
- RESET généré par le logiciel.
- RESET de l'adresse illégale.

Lorsque le RESET se produit :

- l'exécution du programme est interrompue.
- les différents registres et les bits d'état sont affectés.
- le compteur programme PC est mit à l'adresse 0.

Durant le signal RS, les entrées données sont en haute impédance.

### 2- Interruption produite par le matériel .

Il existe deux types :

\*Interruption externe : généré par l'une des cinq broches externes qui sont : XINT1, XINT2, XINT3, PDPINT, et NMI.

-les 4 premiers sont masquables, par un bit dédié et par IMR.

-  $\overline{NMI}$  qui est non masquable, les interruptions masquables sont généralement interdites pendant une interruption NMI.

\*Interruption périphérique : ils sont lancé par les modules périphériques intégrés : EV, SPI, SCI, WD/RTI, et ADC.

Elles peuvent être masquées par un bit dédié.

**N.B** : Le tableau suivant nous indique l'ordre de priorité des interruptions matérielles et leur vecteur d'interruption :

### 3-Interruption générée par le logiciel :

Il existe 4 types d'interruptions :

1-INTR : son opérande indique le vecteur d'interruption, cette instruction inhibe toutes les interruptions masquables.

2-NMI : cette instruction nous branche au vecteur d'interruption 24H.

NMI peut être lancé soit par la broche NMI ou l'instruction NMI.

3-TRAP : appelle d'interruption logiciel, elle n'a pas de priorité relative. lors d'une instruction TRAP , le compteur programme incrémenté de 1 est poussé au sommet de la pile matérielle et le contrôle du programme est transmis à l'adresse 22H en mémoire programme qui peut contenir un branchement à l'adresse de la routine de TRAP.

4-l'émulateur trapu : cette interruption peut être générée soit par l'instruction INTR ou TRAP.

## II.8 Mode d'adressage :

Le TMS320F240 utilise 4 modes d'adressage :

### II.8.1 Adressage direct :

Dans ce mode l'adresse de l'opérande est formée par les 7 bits LSBs de DMA concatène avec les 9 bits du registre DP, pour former une adresse mémoire donnée de 16 bits.

Donc on obtient 512 pages, telle que chaque page contient 128 mots.

### II.8.2 Adressage indirect :

Cet adressage se fait en utilisant l'un des huit registres auxiliaires (Arn). Définir préalablement le registre courant en chargeant le pointeur de registre auxiliaire (ARP). Il est ensuite possible d'utiliser le contenu du Arn sélectionné comme adresse de données.

Il existe sept types d'adressages indirects :

- Adressage indirect auto-immunité.
- Adressage indirect auto décrémentation.
- Adressage indirect post-indexé par ajout de AR0.
- Adressage indirect post-indexé par soustraction du registre AR0.
- Simple adressage indirect.
- Adressage avec bit renversé avec incrémentation.
- Adressage avec bit renversé avec décrémentation.

### II.8.3 Adressage immédiat :

La valeur de l'opérande est stockée de manière explicite dans l'instruction. Cette donnée peut être en format court et permet un codage de l'instruction sur un mot, ou en format long qui nécessite deux mots par instruction.

#### II.8.4 Adressage via un registre :

Il est possible de spécifier un opérande par un registre ; par exemple, TREG0 sert de multiplicande, l'ACCU est implicite pour toutes les opérations liées à l'ALU ... etc.

#### II.9 Les dispositifs contrôlés par le processeur maître :

Le processeur maître commande et contrôle les dispositifs d'E/S suivante :

- **Unité ADC :**

Elle contient cinq convertisseurs numériques-analogiques.

Un convertisseur ADC1 multiplexé par quatre canaux analogique (ADCH1...ADCH4), et la sélection du canal est faite par le logiciel [8].

Ce convertisseur a les caractéristiques suivantes :

16 bits de résolution.

$\pm 10V$  de tension d'entrée.

Temps de conversion de 2  $\mu s$ .

$\pm 5mV$  d'erreur offset.

$\pm 0.25\%$  d'erreur de gain.

Taux du signal de bruit  $> 80dB$ .

Quatre convertisseurs analogiques numériques parallèles (ADC2...ADC5) avec un canal pour chacun (ADCH5...ADCH8).

Ces convertisseurs ont les caractéristiques suivantes :

12 bits de résolution.

$\pm 10V$  de tension d'entrée.

Temps de conversion de 800 ns.

$\pm 5mV$  d'erreur offset.

+ 0.5% d'erreur de gain.

Taux du signal de bruit  $> 70dB$ .

- **Unité DAC :**

Le MPC8240 contrôle les 8 convertisseurs numériques-analogiques, qui ont les caractéristiques suivantes :

8 canaux parallèles (DACH1...DACH8).

16 bit de résolution.

$\pm 10V$  de tension de sortie.

$\pm$  Temps de conversion maximum de 10 $\mu s$ .

$\pm 5\text{mV}$  d'erreur offset.

$\pm 0.5\%$  d'erreur de gain.

Taux du signal de bruit  $> 80\text{dB}$ .

L'unité DAC a 2 modes de fonctionnement :

1-mode transparent :

La valeur convertie est mit en sortie immédiatement.

2-mode verrouillé :

La valeur convertie est en sortie après un signal d'échantillonnage, ce mode n'est pas supporté par RTI.

- **Unité d'E/S numérique :**

Cette unité est contrôlée par le processeur maître MPC8240 avec les caractéristiques suivantes :

--20 bit d'E/S (IO0,...,IO19)

--Leur direction est sélectionnée par le logiciel individuellement.

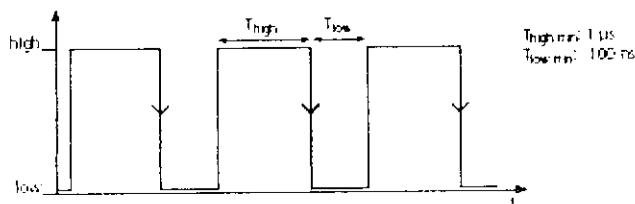
--Courant de sortie maximum  $\pm 5\text{mA}$ .

--Technologie TTL.

**N.B :** Les broches IO16...IO19 peuvent être utilisés comme entrée des signaux d'interruption dans la carte DS1104, mais pour qu'ils soient reconnues par le contrôleur d'interruption :

-le signal doit conserver son état haut pendant au moins  $1\mu\text{s}$ , et des que le front descendant ce produit l'interruption est validé.

-le signal doit conserver son état bas pendant au moins  $100\text{ns}$  après le front, comme le montre la figure suivante :



- **Interface décodeur incrémentale :**

Le maître PPC contrôle cette unité qui a les caractéristiques suivantes :

--Un compteur de position de 24 bits.

--1.65 MHz de fréquence qui cadence le compteur.

--Une alimentation pour encodeur.

--2 canaux d'entrée :

Chacun contient 6 signaux :  $\text{PHI0}(X)$ ,  $\text{PHI90}(X)$ ,  $\text{IDX}(X)$ ,  $\overline{\text{PHI0}(X)}$ ,  $\overline{\text{PHI90}(X)}$ ,  $\overline{\text{IDX}(X)}$ .

**NOTE :**

X indique le numéro de l'encodeur courant.

2 modes :

--mode TTL : les broches inversées doivent être non connectable.

--mode RS422 : en fait la différence entre le signal et son inverse, par exemple :  $\text{PHI0} - \overline{\text{PHI0}}$ .

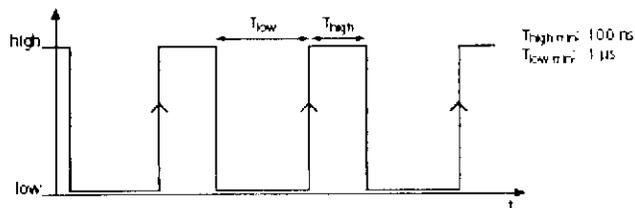
Les encodeurs incrémentales fournissent deux signaux  $\text{PHI0}$  et  $\text{PHI90}$  et le signal d'incrément  $\text{IDX}$ , la paire  $\text{PHI0}$  et  $\text{PHI90}$  sont décalés de  $90^\circ$ .

Les signaux  $\text{IDX}(1)$  et  $\text{IDX}(2)$  sont utilisés comme signaux d'interruption au front montant.

Pour qu'ils soient reconnus par le contrôleur d'interruption :

--le niveau bas doit être conservé pendant au moins  $1\mu\text{s}$ , et au front montant l'interruption est activé

--le signal doit conserver son état haut pendant au moins  $100\text{ ns}$  comme le montre la figure suivante :



• **Interface série :**

La DS1104 contient une interface série (UART) pour permettre la communication asynchrone avec les dispositifs externes.

L'UART peut être configuré en :

-Mode de transmission RS232 : la vitesse de transfert des données est de  $115.2\text{ K Baud}$ .

-Mode de transmission RS422 ou RS485 : la vitesse de transfert des données est de  $1\text{ M Baud}$ .

Ses broches sont :

--RXD,  $\overline{\text{RXD}}$ .

--TXD,  $\overline{\text{TXD}}$ .

--RTS,  $\overline{\text{RTS}}$ .

--CTS,  $\overline{\text{CTS}}$ .

--DCD, DTR, DSR.

La donnée à transférer est initié par le bit START (LSB), le bit de parité à plusieurs modes :

--parité paire.

--parité impaire.

-bit de parité forcé à 0 ou 1.

Suivi de 1 ou 1.5 ou 2 bits Stop.

L'UART fournit une interruption hardware, qui est causée :

-lorsque le nombre d'octet dans le buffer de réception atteint un seuil indiqué.

-buffer de transmission est vide.

-mode ligne d'interruption.

-mode modem d'interruption.

- **Mémoire :**

La carte est équipée de 2 mémoires :

1-mémoire globale :

-mémoire DRAM (SDRAM) synchrone 32 Moctet pour les applications et les données

-mémoire cache.

2-mémoire flash de 8 Moctet avec 4 block de 2 Moctet chacun, telle que :

-6.5 Moctet peuvent être utilisés pour les applications utilisateur.

-1.5 Moctet sont réservés pour les micro-logiciels d'amorçage.

**NB :** Les unités fournies par le DSP esclave ont été citées et détaillées auparavant.

## Conclusion

Nous avons décrit dans ce présent chapitre, les éléments de base constituant la partie hardware (matérielle ) de notre carte de commande, en détaillant le processeur slave TMS320F240 qui est responsable de la génération des signaux de commande MLI qui fera l'objet de notre application au chapitre IV.

La carte DS 1104 avec ses différents périphériques est un vrai module de contrôle et de commande.

# Chapitre:III

---

## Présentation et mise en oeuvre du logiciel

## Introduction :

Le logiciel dSPACE est une interface entre l'utilisateur et la carte ds1104, car il nous facilite l'accès à cette carte directement, pour la programmer. Ce logiciel a un avantage, qui consiste à s'adapter à n'importe quelle carte dSPACE dsxxxx.

Avant l'installation du logiciel de la carte DS1104, on doit d'abord installer :

- Le compilateur C **Microtec power PC** :

Ce compilateur est exigé pour la version de notre kit [9].

- Matlab, simulink et Real-Time Workshop.

Pour l'installation du logiciel il faut suivre les consignes suivantes :

1-Insérer le CD-dSPACE dans le lecteur CD.

2-Insérer la clé-disque dans le lecteur disquette.

3-choisir le lecteur CD-ROM et exécuter le fichier CDsetup.exe

4-fermez tout les programmes de fonctionnement en incluant l'explorateur Windows avant de Continuer l'installation.

5-suivre les instructions provenant du programme d'installation.

6-retirer la clé et le CD-dSPACE.

7-relancer le système d'exploitation.

8-sous Windows NT 4.0 ou Windows 2000, entrez comme un utilisateur avec l'administrateur après réinitialisation.

9-vous pouvez télécharger et installer les nouveaux logiciels de dSPACE disponibles dans le site Internet de dSPACE [16].

Le logiciel de notre carte se compose :

-ControlDesk.

-Interface en temps réel (RTI).

-MLIB\MTRACE.

**N.B** : Pour activer la licence; branchez la clé matérielle dans le port parallèle du PC.

### III-1-Utilisation de ControlDesk [4] :

#### Introduction :

Le logiciel **ControlDesk** est établi pour des expériences logicielles. Il fournit toutes les fonctions pour contrôler, surveiller et automatiser les expériences, il rend ainsi le développement des contrôleurs plus efficace.

ControlDesk est divisé en 3 parties :

- ControlDesk standard.
- ControlDesk multiprocesseur d'extension.
- ControlDesk qui teste l'automatisation.

Pour travailler avec ControlDesk il faut que le PC ait les propriétés suivantes:

-système d'exploitation:

\*Windows 98/ME.

\*Windows NT 4.0 ou Windows 2000.

-la fréquence d'horloge est au moins de 450 MHz.

-la mémoire est de 128 M octet.

-pour l'installation de ControlDesk, on a besoin de 130 M octet d'espace dans le disque dur.

### III-1-1-Principe de fonctionnement :

Après l'installation du logiciel DSPACE par la procédure qu'on a vu précédemment, ControlDesk s'installe automatiquement.

Pour son démarrage, on procède comme suit :

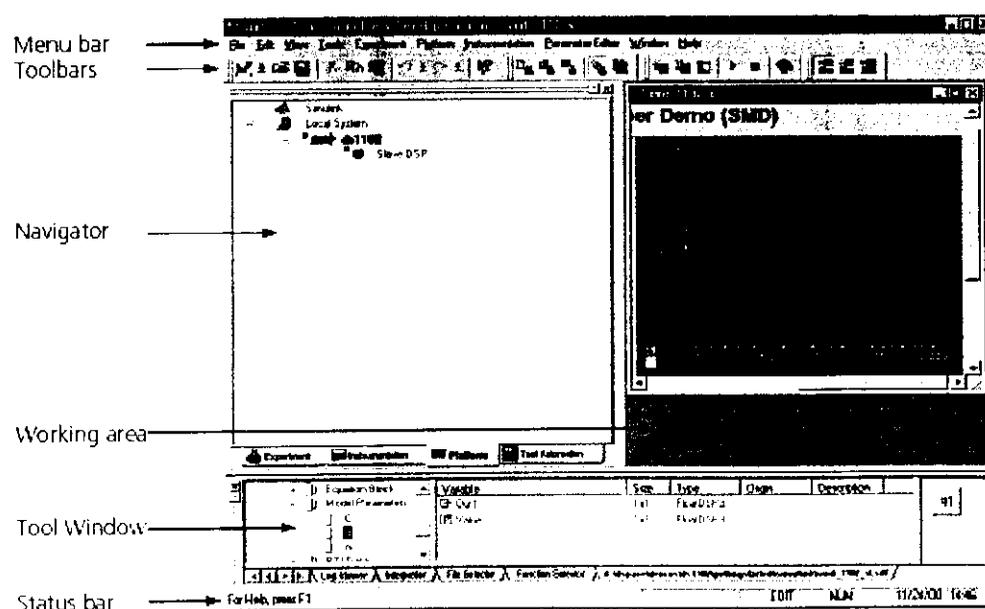
-Dans le menu Démarrer sélectionner :

**programmes-dspace Tools-ControlDesk.**

Alors que pour fermer **ControlDesk** :

-Dans le menu **File**, sélectionner **Exit**.

**III-1-1-1-Les éléments de base de ControlDesk :** En ouvrant ControlDesk, une fenêtre qui contient tout les éléments de base de celui-ci s'ouvre :



Les éléments de base sont :

#### **III-1-1-1- menu bar :**

A travers la barre du menu, on a accès aux différentes fonctions et commandes qui travaille avec ControlDesk.

Les menus sont : file, edit, view, tools ,...,help.

#### **III-1-1-1-2-Toolbars :**

Il est utilisé pour accéder rapidement aux commandes les plus utilisées. Pour le faire apparaître :

-Sélectionner **View-Toolbar** dans le menu.

Il fournit les commandes suivantes :

New, open, save, copy, past ...etc.

#### **III-1-1-1-3-Navigateur:**

Le Navigateur structure les fonctions pour gérer la plateforme ( une plateforme est le nom de la carte qu'on utilise), traite les fichiers, construit les panneaux d'instrumentation, et crée des taches automatiques.

Il se décompose en :

##### **a-Navigateur d'Experiencie (Expiriment Navigator):**

Il affiche tous les fichiers (par exemple, panneau d'instrumentation, positionnement de paramètre, et les donnée de référence) appartenant a l'expérience qui est ouverte, et fournit les fonction pour traiter les expériences et ses composantes.

##### **b-Navigateur d'Instrumentation (Instrumentation Navigator):**

Il permet de construire les panneaux d'instrumentation pour commander et contrôler les variables du model d'une part, et d'autre part de capturer les résultats de l'expérience.

##### **c-Navigateur de plateforme (Platform Navigator):**

Il affiche toute les plateformes enregistrés dans notre système et fournit les fonctions pour enregistrer la plateforme de travail, et traiter les applications telle que charger, démarrer, et stopper l'application.

##### **d-Navigateur de test d'Automatisation (Test automation Navigator) :**

Il fournit toute les fonctions requises pour le test avancé et les taches d'automatisation.

#### **III-1-1-1-4-Zone de travaille (working area) :**

Cette zone est utilisée pour afficher et édité l'expérience, pour créer un layout et une connexion de données dans le cas ou en veut construire un panneaux d'instrumentation.

**III-1-1-1-5-Tool window :**

Elle se décompose:

\*Log viewer (Visualisateur du logarithme naturel):

Log viewer affiche les messages générés par ControlDesk ou de la carte indiquant l'état de fonctionnement et le code d'erreurs.

\*Interpréteur :

L'interpréteur PYTHON traite les commandes PYTHON et les séquences de ControlDesk et teste l'automatisation.

\*File selector :

Ce fichier nous permet de sélectionner et télécharger les applications dont l'extension est (APL,AXP,DDS,M,MDL,OBJ,PPC,SDF) dans la carte.

Pour l'afficher, il faut choisir **Platform Navigator**, et le **File Selector** s'ouvre automatiquement.

\*Function selector :

Il affiche les groupes et les fonctions disponibles dans les modules PYTHON et nous permet de produire des appels de fonction.

**III-1-1-1-6-mode bar (status bar) :**

Cette bar affiche des textes explicatifs des icônes de commande de toolbar ou menu bar.

**III-1-1-2-Affichage des éléments de base de Controldesk :**

On peut sélectionner ou désélectionner les éléments suivants par les fonctions qui se trouvent dans **View** de la bar de menu :

-Toolbars.

-Controlbars.

-Navigator.

-Tool window.

-Status bar.

**III-1-1-3-Fichier de traitement :**

ControlDesk utilise certains dispositifs spéciaux et techniques standard pour les fichiers de traitement.

ControlDesk peut traiter différents types de fichier, qui sont spécifiques par leurs extensions.

Il y'a plusieurs types de fichiers qui sont utilisés par ControlDesk pour diriger les informations et les applications :

- **fichier.CDX (controledesk experiment files)**

Ces fichiers contiennent les liens avec tous les fichiers relatifs à l'expérience.

- **fichier.CON (connection files)**

Ils décrivent la connection entre les instruments et les variables

- **fichier.LAY (fichier layout)**

Ils contiennent les information sur les instruments, taille, position attribuer sur le panneau d'instrumentation, mais il faut noter qu'il n'y a pas de connexion entre les instruments et les variables.

- **fichier.MAT**

Il est utilisé pour sauvegarder les données qui sont capturé, ou éditer les tables de données.

- **fichier.MDL**

C'est un fichier crée par simulink.

- **fichier.OBJ**

C'est un fichier exécutable par le processeur en temp réel dans la carte ds1003, ds1102, et le DSP esclave (ds1103, ds1102,et ds2201).

- **fichier.AXP**

Il est utilisé par le ds1004 et le fichier.PPC par ds1103, ds1005,et ds1401.

- **fichier.PAR (parameter files)**

Il contient la description, les types, et les valeurs des paramètres placés dans le model de simulink.

- **fichier.SDF**

Ce fichier décrit les fichiers chargés individuellement.

Il est généré automatiquement lorsqu'on construit le fichier.TRC avec RTI ou la simulation de simulink.

On peut utiliser l'éditeur de fichier SDF pour générer et éditer le fichier.SDF.

- **fichier.TRC**

Il fournit les informations des variables disponibles et comment elles sont groupées.

#### **III-1-1-4-Création de Nouveaux Fichiers :**

En créant un nouveau fichier, vous devez indiquer le type du fichier à créer qui à une extension de type : TXT, PY, C, H, SE, et LAY. Pour le créer, il faut choisir **file-new** dans la barre de menu de ControlDesk.

Pour chacun de ces types de fichier, ControlDesk ouvre une nouvelle fenêtre dans l'emplacement de travail et appelle l'outil correspondant au type choisi de fichier.

D'autres types de fichier sont créés et sauvegardés avec les outils associés:

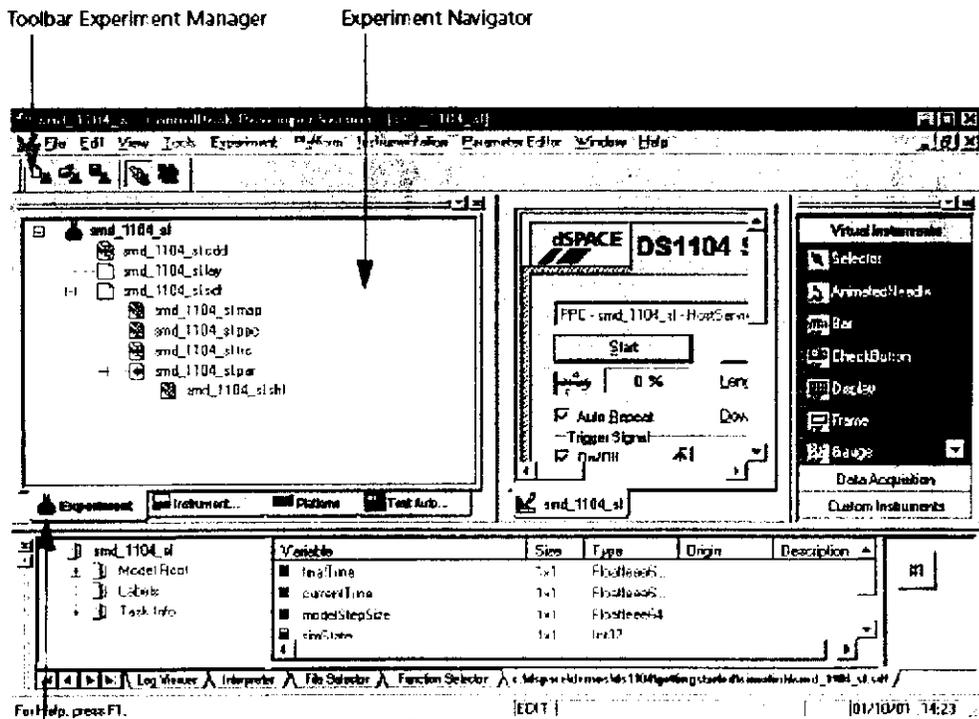
CDX	Experiment Manager	Read/Write
CON	Instrumentation Navigator	Read/Write
FLT	Filter Manager of the Variable Manager	Read/Write
PAR	Variable Browser	Read/Write
SDF	System Description File Editor and Simulink/Real-Time Interface	Read/Write
SE	Stimulus Editor	Read/Write
MDL	Simulink Interface	Read/Write
TRC, MAP, OBJ	Simulink (TRC) and Real-Time Interface	Read

### III-1-2-La gestion des expériences :

Le manager d'expérience est un outil de ControlDesk qui contrôle les fichiers appartenant à l'expérience, et fournit les commandes nécessaires pour traiter les expériences.

#### III-1-2-1-Les bases du manager d'expérience :

Pour accéder au manager d'expérience, cliqué sur **experiment** dans la fenêtre du navigateur, on obtient ainsi :



Expérier ent tab of the Navigator

La plus part des commandes du manager d'expérience sont disponibles dans le menu **File**, menu **experiment** ou **Toolbar**.

**III-1-2-2-L'utilisation du navigateur d'expérience :**

Pour travailler avec une expérience, il faut respecter les étapes suivantes :

- 1-Crée une nouvelle expérience en utilisant la bar de menu (**File-New Experiment**), ou modifier les propriétés de l'expérience existante.
- 2-Ajouter les fichiers à l'expérience, ou retirer les fichiers de l'expérience.
- 3-Sauvegarder, fermer et charger l'expérience.
- 4-archiver l'expérience et ses fichier ou envoyer certain fichier a un destinataire de chemise ou de courrier.
- 5-on peut également vouloir utiliser l'expérience dans une autre plateforme.

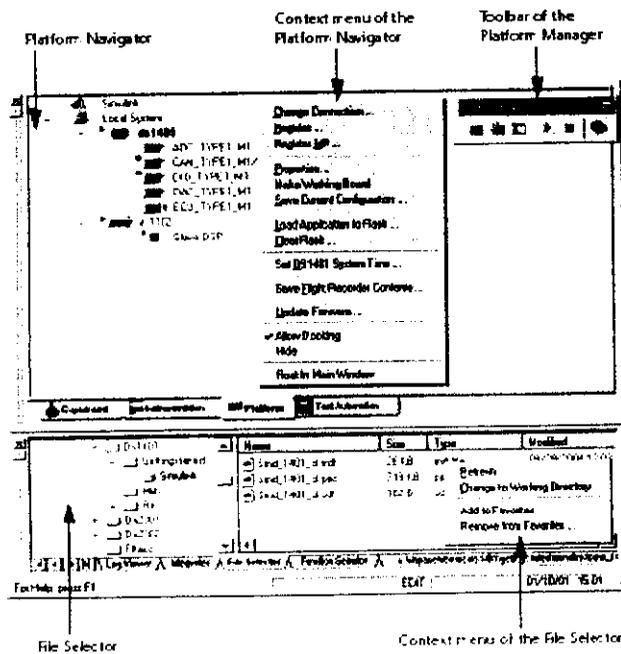
**III-1-3-La gestion des plateformes :**

Le gestionnaire de plateforme fournit les fonctions pour traiter les plateformes installer dans le PC , et les application assigner à cette plateforme.

**III-1-3-1-Les éléments de bases du gestionnaire de plateforme :**

Le Navigateur de Plateforme affiche toute les plateformes enregistrés par ControlDesk, et fournit une fonctionnalité complète et confortable pour la gestion des plateformes. Le File Selector qui est localisé dans Tool Window du ControlDesk nous permet de charger les applications dans la plateforme par drag et drop.

La figure suivante nous donne ces éléments de base :



### III-1-3-1-1-Les commandes du gestionnaire de plateforme :

Le navigateur de plateforme et le **file selector** fournissent les commandes exigées. Certaines de ces commandes sont disponibles dans le menu de plateforme et la barre d'outils du gestionnaire de plateforme (cette barre d'outil est disponible lorsque on sélectionne **-plateforme manager-** dans le menu **View-Toolbars** ).

### III-1-3-2-Enregistrement des Plateformes :

Une des fonctions du navigateur de plateforme est l'enregistrement des plateformes utilisées. Pour enregistrer un panneau de processeur, il faut suivre les étapes suivantes :

1-Du menu bar, choisir **Platform-Initialization-Register**

2-Dans l'interface **Register Board** , sélectionner le type de carte installée (DS1104).

3-Ecrire l'adresse de base d'E/S sélectionnée dans la zone **Port address** en hexadécimal.

4-Dans l'interface **Register Board** cliquer sur **Register** pour compléter l'enregistrement et retourner à la plateforme du navigateur [7].

### III-1-3-3-Creation du fichier SDF :

Le fichier SDF est créé par RTI et l'éditeur de fichier de description du système qui nous permet de créer et éditer les données spécifiques à l'application pour n'importe quelle plateforme.

Pour créer un fichier SDF, il y a deux méthodes :

#### III-1-3-3-1-Editeur de fichier de description de système (SDF) :

1-Dans la barre menu choisir **Tools-SDF File Editor**, ou une fenêtre apparaît.

2-Du **Select board type**, choisir la plateforme désirée et cliquer sur **Add Selected** .

3-Dans **Single board** , écrire le nom de la plateforme, l'application qui est utilisée par le processeur, et cliquer **OK**.

#### III-1-3-3-2-Création automatique :

Des que le modèle est prêt, on clique sur **Build** et le fichier SDF est créé automatiquement.

### III-1-3-4-Chargement de l'application dans le processeur :

Pour télécharger et commencer l'application en temps réel :

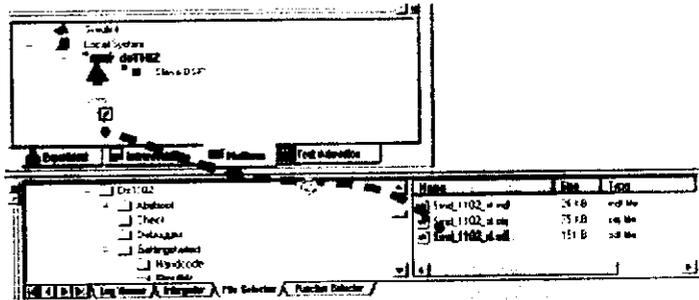
1-cliquer sur le navigateur de plateforme, **File Selector** est affiché automatiquement.

2-cliquer sur variable browser

3-cliquer sur le fichier SDF désiré et le traîner à l'icône plateforme dans le navigateur de plateforme.

L'application est chargée en mémoire globale de la carte dSPACE et commence à être exécuté par le processeur.

La figure suivante se montre :



Pour stopper l'exécution, par la carte de l'application :

1-dans le navigateur de plateforme, sélectionner le panneau.

2-de la bar de menu, choisir **Platform-Application-Stop Real-Time Processor**.

**Notel :** Pour télécharger l'application en temps réel en mémoire flash et débiter le RTP (application en temps réel) :

1-Dans le Navigateur de Plateforme, sélectionner le panneau ds1104.

2-Du bouton droit de la souris, choisir **Load Application to flash**.

3-Choisir l'application (fichiers SDF ou PPC), dans la fenêtre de dialogue :

**Load Application to RTP.**

**Note 2 :** Pour effacer une application de la mémoire flash :

1- Dans le Navigateur de Plateforme, sélectionner le panneau ds1104.

2-stopper l'exécution de l'application en temps réel par la carte.

3-Dans le menu de l'icône du panneau, choisir **Clear Flash**.

4-Dans l'interface graphique **Clear Flash EPROM**, on a deux options pour effacer la mémoire flash :

-**Clear complete flash memory** pour effacer toute la mémoire flash

-**Clear flash application memory** pour efface uniquement l'application.

5-cliquer **Clear Memory** pour effacer la mémoire flash en accord avec le point 4.

### III-1-4-La construction des panneaux d'instrumentation :

ControlDesk Standard fournit un ensemble d'instruments puissants. Ils sont construits pour contrôler et commander les variables de l'application en temps réel, et pour l'affichage des données capturées.

Ces instruments peuvent être arrangés librement en une ou plusieurs fenêtres. Elles sont appelées Layout Window, chaque Instrument connecté à une variable est appelé **Instrument Vertuel**.

### III-1-4-1-Outil d'instrumentation :

Les outils d'instrumentation sont :

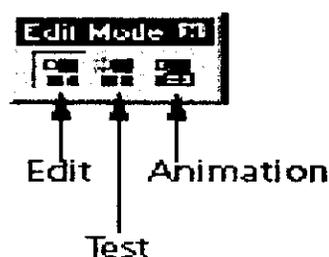
-Navigateur d'instrumentation ; Panneaux d'instruments ; Sélecteur d'instruments et La fenêtre Layout.

#### III-1-4-1-1-Navigateur d'instrumentation :

Le Navigateur d'instrumentation affiche la hiérarchie de tous les panneaux d'instruments ouverts et leurs instruments.

Plusieurs commandes d'instrumentation sont disponibles à travers le menu Instrumentation, et la barre d'outil qui est disponible pour déterminer le mode du panneau d'instrument. La figure ci-contre

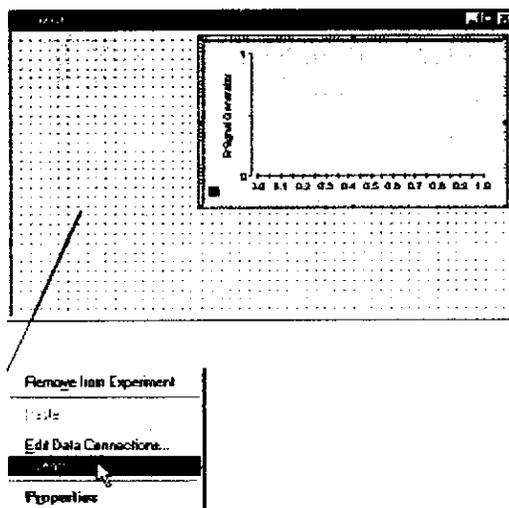
nous montre l'interface graphique des différents modes :



#### III-1-4-1-2-Panneaux d'instruments :

La fenêtre Layout est utilisée pour construire le panneau d'instrumentation. On peut créer des instruments dans cette zone selon nos besoins d'applications et connecter les variables à l'application réelle ou à la simulation de simulink.

La figure suivante nous montre une fenêtre layout et son menu :



### III-1-4-1-3-Selecteur d'instruments :

Le sélecteur d'instruments permet un accès à tous les instruments disponibles, ils sont divisés en trois groupes :

#### a-Les instruments virtuels :

Ils sont utilisés pour afficher les valeurs des variables connectés, ou pour écrire des valeurs de la plateforme de simulation.

Le tableau suivant nous indique les différents instruments et leurs rôles :

	Animator\Needle	To display the value of a connected numeric variable by a needle deflection.
	Bar	To display the value of a variable by a bar deflection.
	CheckBox	To display whether the value of a connected variable or instrument matches predefined values or to write a predefined value to a connected variable or instrument.
	Display	To display numerical values with a predefined format.
	Frame	To arrange instruments on the layout.
	Gauge	To display the value of a connected numeric variable by a needle deflection on a circular scale.
	InvisibleSwitch	To simulate the different buttons (On/Off button, Push button, Check button) with a transparent background.
	Knob	To set the value of the connected numeric variable or display the value of the connected variable on a circular scale.
	Message	To display messages that are related to values of the connected numeric variable.
	MultiStateLED	To display an LED or image that is related to values of the connected numeric variable.
	NumericalInput	To enter or display a value that is transferred to a connected variable or instrument.
	OnOffButton	To set the value of the connected numeric variable or activate/close an event-based layout.
	PushButton	To write a predefined value to a connected variable or instrument and/or start an action, for example to call up a layout.
	RadioButton	To set an option and transfer a specified value to the simulation platform or display the state of a variable.
	Slider	To enter values of a connected variable to the simulation platform or display the values of the variable.
	StaticText	To display explanations or inscriptions on the layout.
	TableEditor	To display and change values of a connected table variable in a chart and a grid view.

#### b-Instrument d'acquisition de données :

Ils sont utilisés pour la capture des données de la plateforme de simulation. Le tableau suivant nous indique les différents instruments et leurs rôles :

	Capture Settings	To control/observe the properties of data captures.
	LogicAnalyzer	To display bit patterns of captured integer variables.
	Plotter	To display the values of several variables relative to different y-axes.
	Template	To combine the functions of the LogicAnalyzer, Plotter and XYPlot instruments.
	XYPlot	To display variables as functions of other variables.

### c-Les instruments personnalisés :

Grâce à ce groupe, on peut spécifier nos instruments ou groupe d'instruments et les sauvegarder dans la librairie.

#### III-1-4-2-Création du panneau d'instruments :

Pour créer une nouvelle fenêtre layout, il faut :

- 1-De la barre de menu, choisir **File-New-layout**
- 2- De la barre de menu, choisir **File-save** ou **File-save As** pour sauvegarder le nouveau panneau d'instruments.

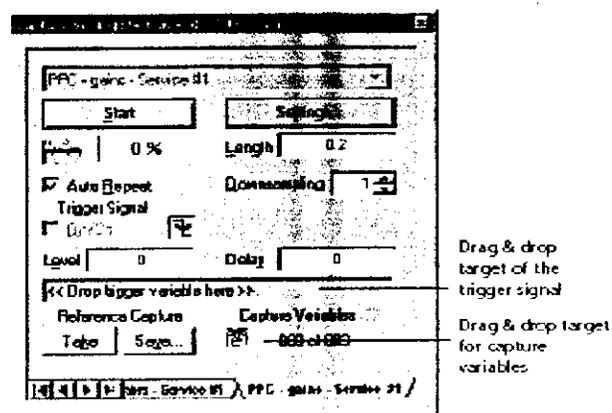
#### III-1-4-3-Connexion de donnée entre les variables et instruments :

Une fois qu'on a créé un nouveau layout, on effectue les étapes suivantes pour pouvoir connecter les variables aux instruments :

- 1-Selectionner le groupe d'instrument du Sélecteur d'instrument.
- 2-choisir l'instrument et cliquer sur son icône
- 3-déssiner un rectangle pour affiche cette instrument
- 4-Selectionner la variable dans –**Variable browser**.
- 5-glisser par la souris (gauche) dans l'instrument sélectionné au préalable.

#### III-1-5-La capture des données :

La fenêtre **capture setting** est utilisée pour le contrôle de la donnée saisis dans ControlDesk, comme le montre la figure suivante :



La saisie de données commence immédiatement après le début de l'animation, sa durée est déterminée en seconde par **Length**. On peut répéter la capture par le **checkbox Auto Repeat**.

Le signal de déclenchement (descendant dans ou montant) est configuré par le **Level**, Il nous permet de capturer la donnée à chaque fois que le front arrive, telle que la durée entre deux fronts est déterminée par **Length**.

## III-2-Utilisation de RTI [44]:

### Introduction :

RTI est une interface entre MATLAB \ Simulink et la carte DS1104. Il permet la communication entre l'environnement Simulink, Real Time Workshop, et l'environnement matériel.

Il faut noter que RTI est créé après l'installation du logiciel DSPACE.

### III-2-1-Accès aux blocks de RTI :

Avant d'accéder aux blocks RTI, il faut choisir la plateforme RTIxxxx de travail courant, cela s'effectue après l'ouverture de Matlab (R11 ou R12 ou R13), qui fait apparaître une interface graphique, qui permet le choix de la plateforme désiré, qui doit correspondre au matériel (pour une carte ds1104, il faut choisir la plateforme RTI1104).

Il existe deux moyens pour accéder aux blocks RTI :

-Simulink library browser.

-Librairie RTI.

### III-2-1-1-Simulink library browser :

Pour y accéder il faut :

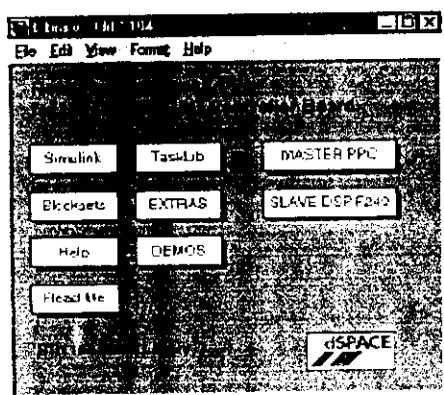
-Ouvrir Simulink library browser par la commande **simulink** de Matlab ou par le bouton de droite de la fenêtre de Matlab.

-Sélectionner dans la liste DSPACE RTI1104 pour afficher les différents blocks.

### III-2-1-2-La librairie RTI :

Il est recommandé d'utiliser cette méthode, pour ouvrir la fenêtre library **rtilib 1104**, il faut pour cela écrire **rtilib** dans le prompt de Matlab (R11,R12,R13).

La fenêtre des blocks de RTI (**rtilib1104**) est affichée comme le montre la figure suivante :



Les composants de rtilib 1104 sont :

#### III-2-1-2-1-Simulink :

Il appelle les blocks de la librairie de simulink standard.

#### III-2-1-2-2-Blocksets :

Il inclus les blocks optionnelles de RTI pour le ds1104.

#### III-2-1-2-3-Help :

Il affiche les informations de référence.

#### III-2-1-2-4-Readme :

Il affiche les informations des notes importantes.

#### III-2-1-2-5-Tasklib :

Il offre des blocks RTI d'interruptions.

#### III-2-1-2-6-EXTRAS :

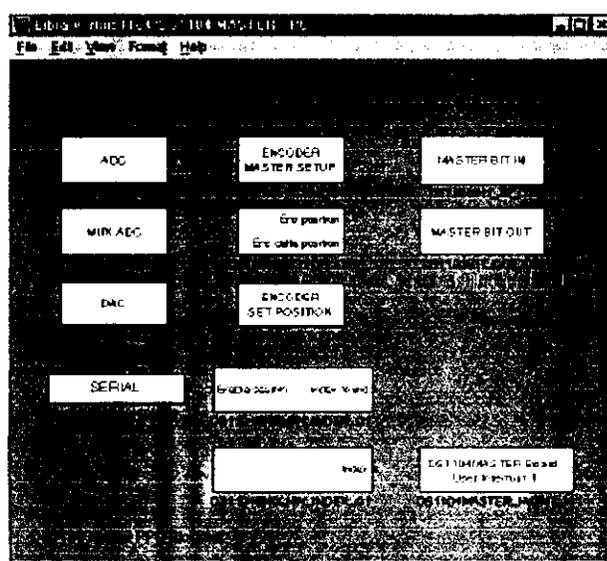
Il offre des blocks RTI pour des buts spéciaux, par exemple le code service de ControlDesk.

#### III-2-1-2-7-Demos :

Il nous montre des exemples du modèle.

#### III-2-1-2-8-Master PPC :

Après avoir cliqué sur le bouton Master PPC, la fenêtre rtilib 1104\DS1104 Master PPC est affichée comme le montre la figure suivante :

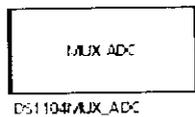


Elle contient les blocks d'E/S servis par le processeur maître Power PC 603e .  
 Les blocks principaux qui permettent le contrôle du processeur maître sont :

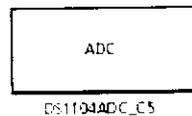
**a-Block ADC :**

Les caractéristiques de cette unité ont été déjà vues dans le chapitre précédent, on a deux type de blocks comme le montre la figure suivante, qui permettent d'activer ou désactiver les cinq unités ADC.

**DS1104MUX\_ADC**



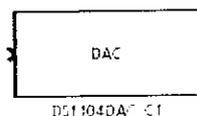
**DS1104ADC\_Cx**



Le premier block permet de sélectionner le canal ou les canaux (quatre ADCHi), qui vont permettre aux signaux analogiques d'être converti en numérique, par un seul ADC.  
 Alors que le deuxième permet le choix de l'un des quatre autres canaux (ADCH5,...,ADCH8), lier chacun à un convertisseur ADC.

**b-Block DAC :**

**DS1104DAC\_Cx**



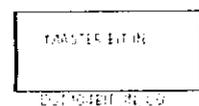
Même chose du point de vue caractéristique. Il permet d'activer ou désactiver les huit DACs.

**c-Block de l'unité d'E/S numérique :**

Il existe deux types de blocks :

**-block d'entrée numérique :**

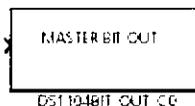
**DS1104BIT\_IN Cx**



Ce block permet de lire un des 20 bits de l'unité d'E/S numérique.

**-block de sortie numérique :**

Ce block permet d'écrire sur un des 20 bits en sortie de l'unité d'E/S numérique.

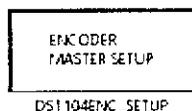
**DS1104BIT\_OUT\_Cx**

Il faut noter qu'on va les utiliser pour faire sortir les signaux MLI 2 et 3 niveaux, dans le chapitre suivant.

**d-Block de codeur incrémental :**

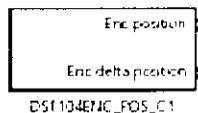
La librairie du maître contient plusieurs blocks pour programmer l'interface de codeur incrémental.

Les blocks les plus importants sont les suivants :

**DS1104ENC\_SETUP**

Dans le cas où on veut utiliser les autres blocks de l'encodeur, le block suivant doit être placé dans notre modèle.

Ce block nous permet de choisir le type de signal qu'on a vu dans le chapitre précédent.

**-block de position :****DS1104ENC\_POS\_Cx**

Il est utilisé pour lire la position et la vitesse de l'un des deux canaux d'encodeur.

**-block de position :****DS1104ENC\_SET\_POS\_Cx**

Il est utilisé pour écrire dans le compteur de l'un des deux canaux.

**e-Block d'interface série :**

Après un double cliquée sur le bouton **Serial**, la fenêtre suivante s'ouvre :

**a-DS 1104 sl\_dsp\_pwm:**

Il est utilisé pour générer des signaux MLI monophasés avec un coefficient d'utilisation variable.

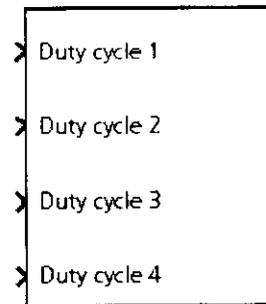
On peut le configurer en deux modes :

1-mode asymétrique :

L'impulsion commence au début de la période, la fréquence de la MLI varie entre 2.5Hz et 10MHz.

2-mode symétrique :

La fréquence de la MLI varie entre 1.25 Hz et 5MHz.

**DS1104SL\_DSP\_PWM**

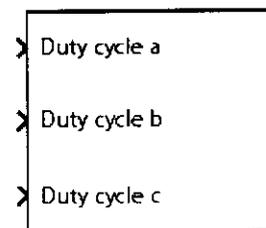
DS1104SL\_DSP\_PWM

**b-block triphasé :**

Il génère une MLI triphasée avec une zone morte et un coefficient d'utilisation variable.

La fréquence varie entre 1.25 Hz et 5MHz.

**NOTE :** Pour accéder aux propriétés de n'importe quel block, il faut double cliquer sur le block correspondant.

**DS1104SL\_DSP\_PWM3**

DS1104SL\_DSP\_PWM3

**III-2-2-Construction et téléchargement du modèle :**

Une fois le modèle est prés, on effectue les étapes suivantes transformant le modèle.mdl en un modèle.SDF :

-Choisir Tools-Real-Time Workshop-option

-Choisir Solver pour le configurer

-choisir Real Time Workshop :

--changer la configuration de : Système target file : rti1104.tlc.

-appuyer sur le bouton build pour compiler notre modèle en un model exécutable par la carte ds1104 qui est le model.SDF.

### III-2-2-1-Les phases de la construction :

L'interface RTI, qui utilise Real-Time Workshop génère le fichier SDF, durant cette phase de construction, plusieurs fichiers intermédiaires sont générés.

La construction du processus est devisée en deux phases :

- La phase de génération de code.
- La phase de compilation.

#### III-2-2-1-1-La phase de génération de code :

Dans cette phase, tout le code C requis pour l'application en temps réel est généré.

La commande `make_rti` commence, et exécute les étapes suivantes séquentiellement :

- Real Time Workshop est appelé pour généré un fichier intermédiaire `model.rtw` du modèle simulink, ce fichier est utilisé comme base pour la génération de code par Target Language Compiler (compilateur de langage cible), généré par RTI et RTW.
- Le compilateur de langage cible (TLC) est appelé, il lit le fichier `model.rtw` pour générer le code C du modèle.
- Le `make file` appelle le `model.mk`, qui est crée par RTI template `make file`.

#### III-2-2-1-2-La phase de compilation :

La source C est traduite par le compilateur en application final:(fichier SDF) qui peut être exécuté par le matériel.

Le fichier `mk` est appelé par l'utilitaire `dsmake` qui appelle, le compilateur Microtec Power PC qui compile notre programme du C en un fichier SDF.

### III-3-Bibliothèque d'interface Matlab-dspace [43]:

La bibliothèque d'interface Matlab-dspace nous donne un accès aux matérielles dspace à partir de l'espace de travaille de Matlab, cette bibliothèque contient les fonction MLIB/MTRACE qui peuvent être appelé de la fenêtre de commande de Matlab ou par les fichier M de Matlab.

MLIB/MTRACE appartient à l'environnement du logiciel de dSPACE, qui contient aussi Matlab, Simulink, Real-Time Workshop, RTI, et ControlDesk.

MLIB/MTRACE fournit les fonction de base pour la lecture et l'écriture de donnée, et autre fonction pour la génération d'interruption, la configuration de l'état du processeur, et fournit des informations sur l'état du processeur.

### **Conclusion :**

On à vu dans ce chapitre, l'utilisation de cet outil formidable qui est le logiciel dSPACE, qui rend la programmation et l'implémentation très facile, ce qui arrange les électrotechnicien qui n'auront plus besoin d'être des experts en programmation, pour commander leurs moteurs.

Cela nous introduit dans le chapitre suivant, qui nous permet de générer une MLI à partir de Matlab \ Simulink.

# Chapitre:IV

---

# Applications

## Introduction :

Dans ce chapitre, et afin de bien comprendre le logiciel, on va implémenter un programme simulink qui génère des signaux de commande MLI deux et trois niveaux (à une et deux porteuses) dans une carte DS1104.

Il faut tout d'abord définir la structure de l'onduleur (onduleur deux et trois niveaux) qui va être attaqué par ces impulsions MLI, ainsi que sa stratégie de commande. Ensuite, on va l'implémenter dans notre carte DS1104. Et enfin, on va récupérer les signaux de sortie à travers le connecteur P1 [8], et les visualiser en utilisant le panneau d'expérimentation de ControlDesk et un oscilloscope.

Afin de générer une source de tension sinusoïdale, différentes stratégies de commande ont été proposées, parmi lesquels, la Modulation de Largeurs d'Impulsions "MLI" (en Anglo-Saxon, Pulse Width Modulation), qui permet d'obtenir une source de tension alternative variable en fréquence et en amplitude.

La Modulation de Largeurs d'Impulsions consiste à adopter une fréquence de commutation supérieure à la fréquence des grandeurs de sortie et à former chaque alternance d'une tension de sortie d'une succession de créneaux de largeurs convenables [10].

L'essor de la Modulation de Largeurs d'Impulsions est lié aux progrès réalisés sur les semi-conducteurs de puissance commandés à l'ouverture et à la fermeture. L'augmentation du nombre de commutations par période entraînerait des pertes excessives si on n'avait pas réussi à réduire les pertes à chacune des commutations [11].

L'intérêt de la M.L.I. est de remédier aux problèmes des harmoniques. Elle présente les avantages suivants:

- ❖ La variation de la fréquence et de l'amplitude de la tension de sortie.
- ❖ L'élimination de certains harmoniques de tension car elle repousse les harmoniques de la tension de sortie vers les fréquences élevées afin de faciliter leur filtrage.
- ❖ La minimisation de l'ondulation du courant et du couple électromagnétique.
- ❖ La réduction du coût du filtre de sortie.

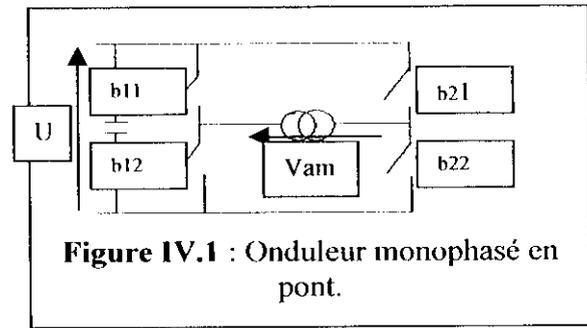
La MLI est caractérisée par deux paramètres :

- L'indice de modulation  $m$  : c'est le rapport de la fréquence  $f_p$  de la porteuse à la fréquence  $f$  de la tension de référence ( $m = f_p/f$ ).
- Le taux de modulation  $r$  : c'est le rapport de l'amplitude de la tension de référence  $V_m$  sur l'amplitude  $U_{pm}$  de la porteuse ( $r = V_m/U_{pm}$ ).

**IV-1- Stratégie de commande d'un onduleur deux niveaux :**

La loi de commande :

La structure d'un onduleur en pont complet monophasée est montrée à la **Figure IV.1** ci-contre.



**Figure IV.1 : Onduleur monophasé en pont.**

La stratégie de commande utilisée est la stratégie comportant une porteuse bipolaire, comparée avec deux signaux de références :

$$\begin{cases} V_{r1} = V_m \cdot \sin(\omega t) \\ V_{r2} = V_m \cdot \sin(\omega t - \pi) \end{cases} \quad (IV.1)$$

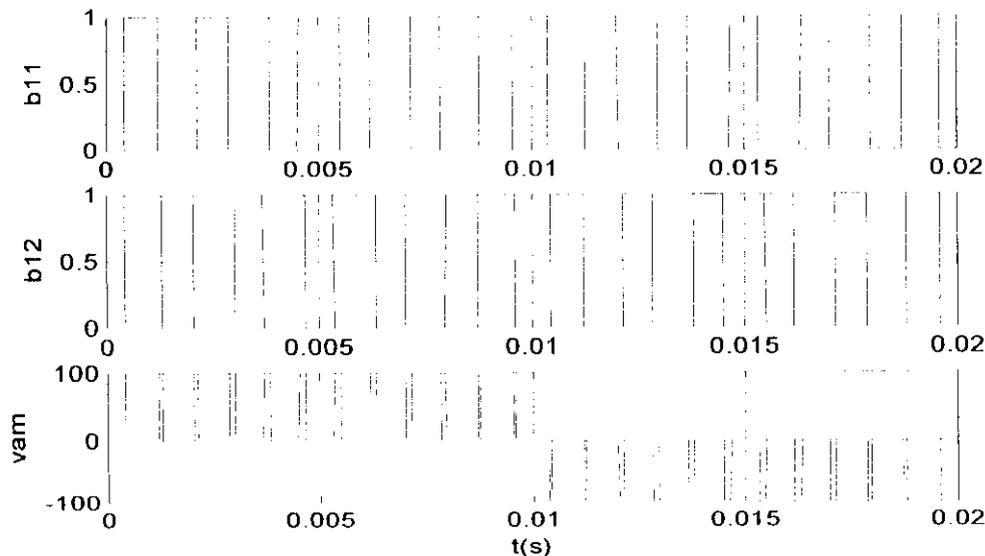
Est telle que :

- On ferme b21 pendant la première demi période, et b22 pendant l'autre demi période.
- On prend b11 et b12 comme interrupteurs de hachage, telle que :

$$\begin{cases} \text{Si } V_{r1} > U_p \Rightarrow b11 = 1 \\ \text{Si } V_{r2} > U_p \Rightarrow b12 = 1 \end{cases} \quad (IV.2)$$

**IV-1-1-Résultat de simulation :**

Les résultats de la simulation des signaux de commande b11 et b12 et de la tension de sortie Vam, pour m=12, r=0.8 et r=0.2 respectivement sont montrées dans les figures (IV.2 et IV.3).



**Figure IV.2 : Signaux de commande et la tension de sortie de l'onduleur deux niveaux en pont complet utilisant la stratégie triangulo-sinusoidale (m=12, r=0.8).**

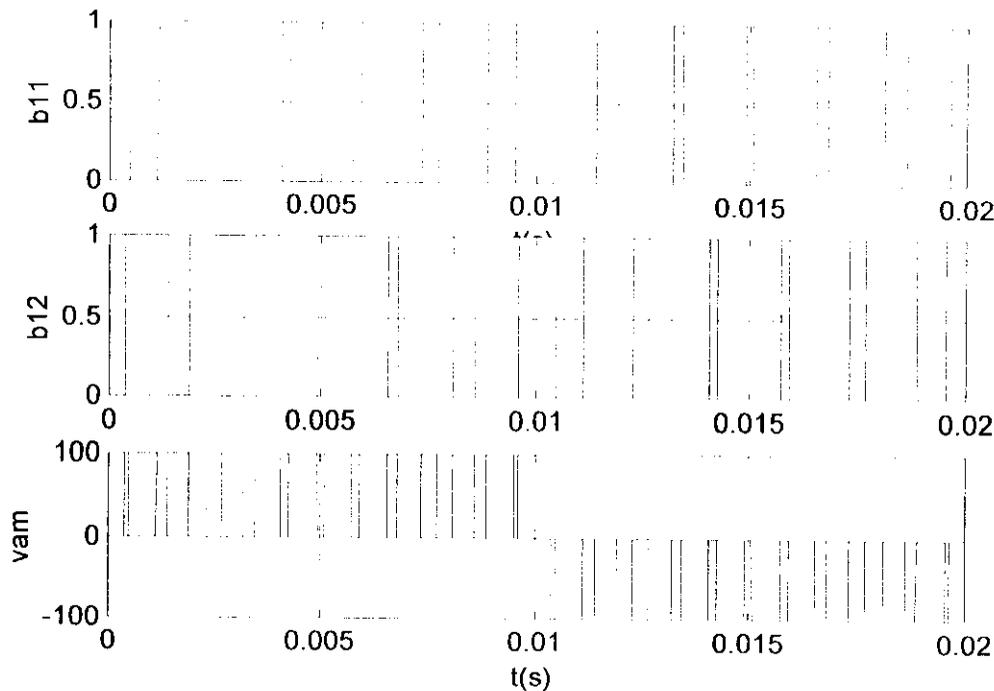


Figure IV.3 : Signaux de commande et la tension de sortie de l'onduleur deux niveaux en pont complet utilisant la stratégie triangulo-sinusoïdale ( $m=12, r=0.2$ ).

IV-1-2-Acquisition des signaux MLI sur la carte dSPACE :

En procédant selon l'organigramme présenté à l'annexe III, on obtient les résultats montrés ci dessous. Ces résultats sont obtenus sous ControlDesk. Les mêmes signaux sont visualisés sur l'oscilloscope.

Ces résultats sont obtenus en prenant  $m=12$  et  $r=0.8$  et  $r=0.2$ .

Note : On fait varier le r directement du panneaux d'instrumentation grâce au slider [7].

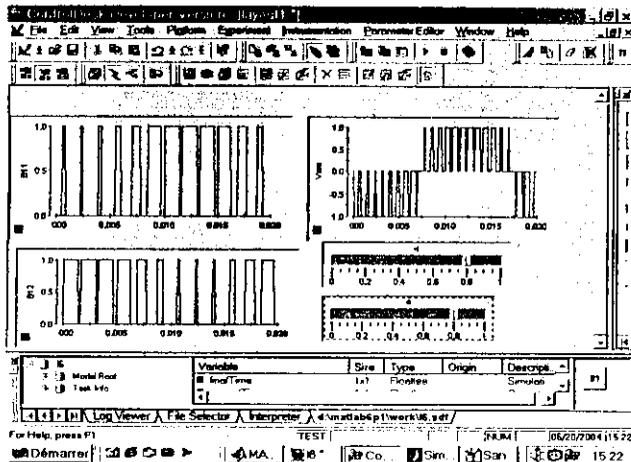


Figure IV.4.1 : MLI d'un onduleur deux niveaux ( $r=0.8$  et  $m=12$ ).

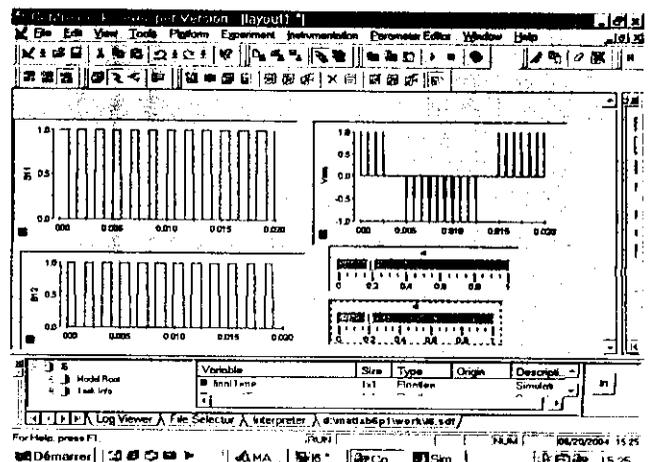
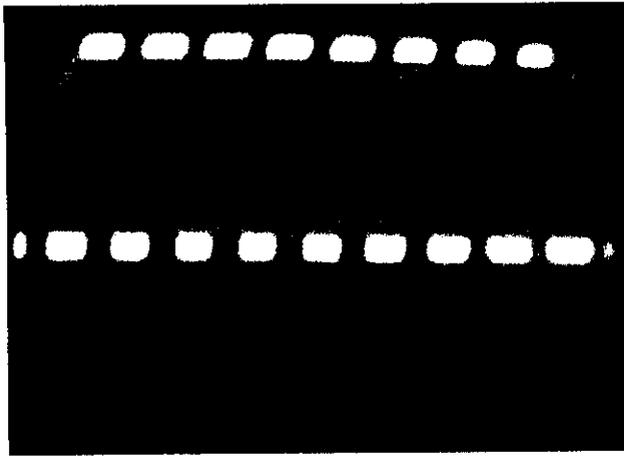
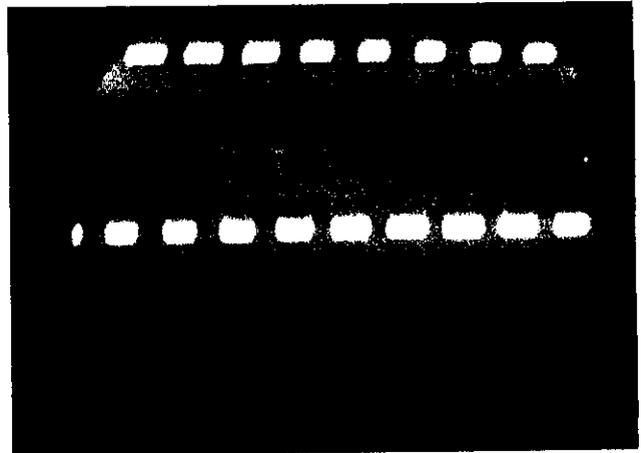


Figure IV.4.2 : MLI d'un onduleur deux niveaux ( $r=0.2$  et  $m=12$ ).

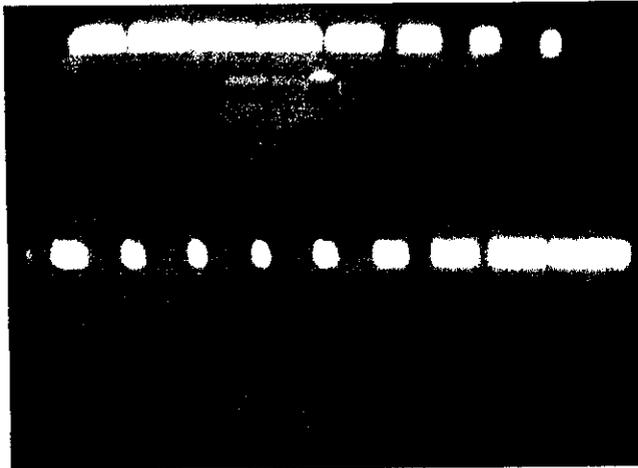
Les figures suivantes représentent les signaux de commande des bases des transistors de l'onduleur deux niveaux (b11 et b12) ainsi que la tension de sortie (Vam).



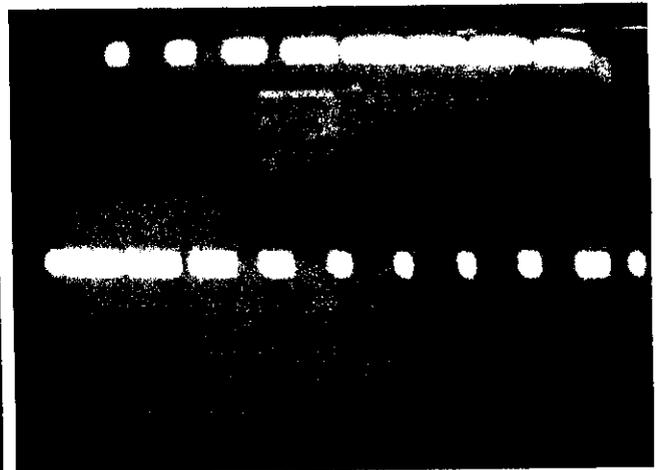
a) Forme de **b11** pour  $m=12$  et  $r=0,8$ .



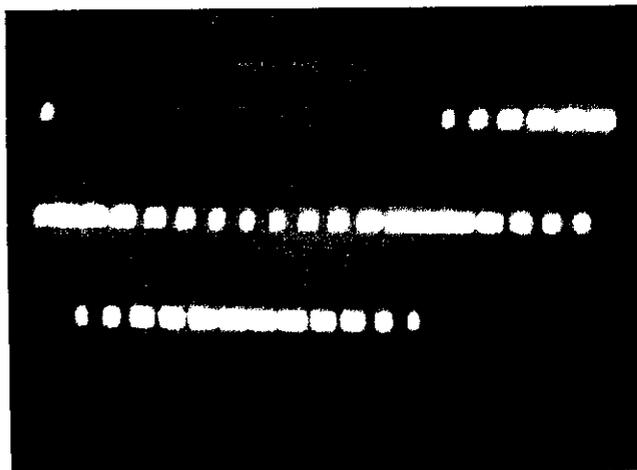
b) Forme de **b12** pour  $m=12$  et  $r=0,8$ .



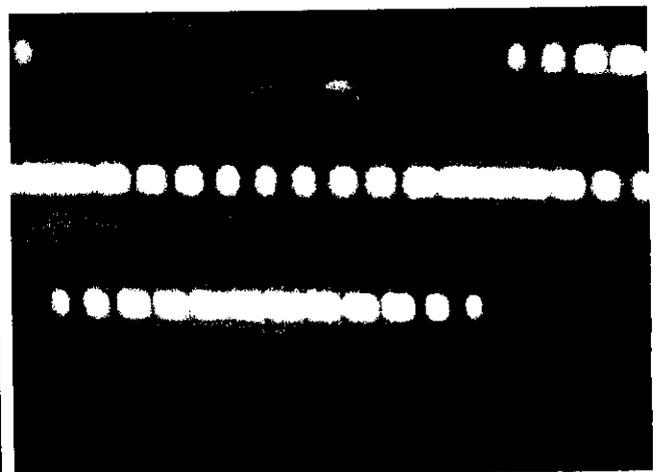
c) Forme de **b11** pour  $m=12$  et  $r=0,2$ .



d) Forme de **b12** pour  $m=12$  et  $r=0,2$ .



e) Forme de **Vam** pour  $m=12$  ;  $r=0,2$ .



f) Forme de **Vam** pour  $m=12$  ;  $r=0,8$ .

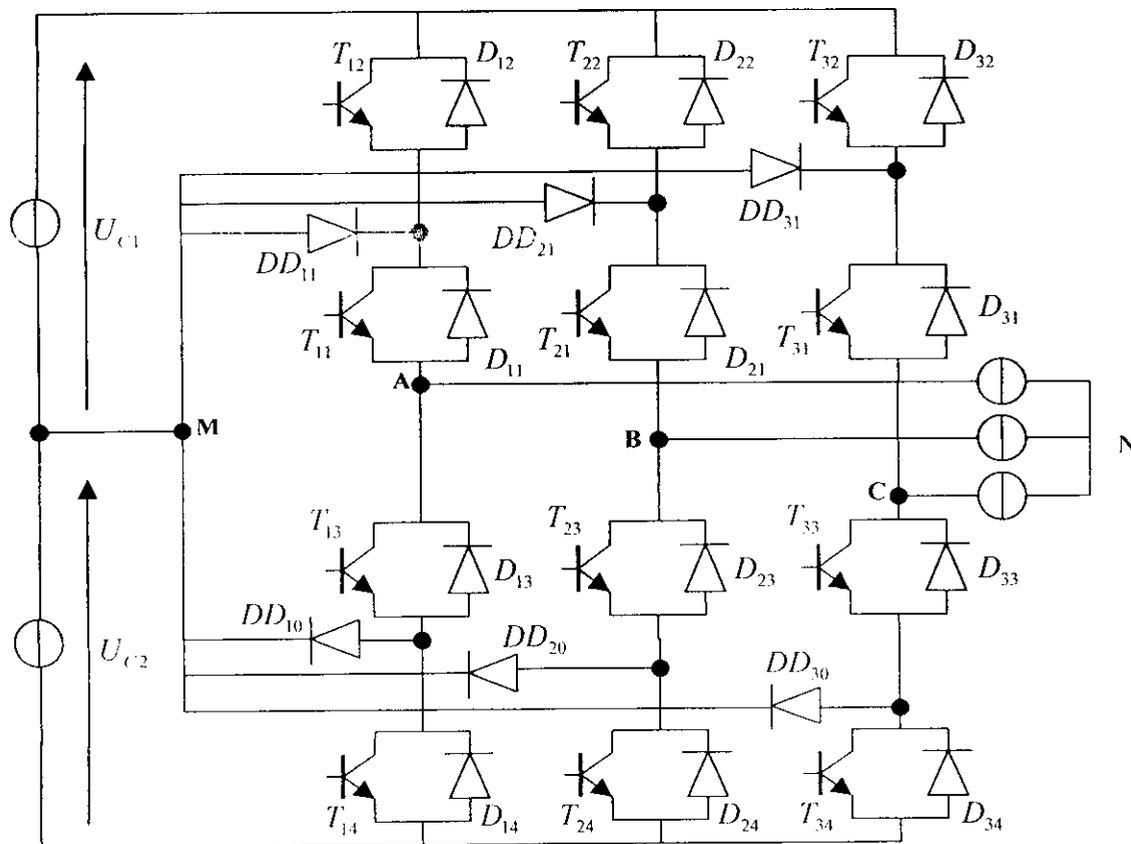
**Figure IV.5** : visualisation des résultats obtenus sur l'oscilloscope.

**Remarques :**

- En diminuant  $r$  (tout en gardant  $m=12$ ), on remarque que la largeur des signaux de commande et des tensions de sortie diminuent.
- A valeurs données de  $m$  ( $m=12$ ) et de  $r$  :  
les formes d'ondes des tensions et des signaux de commande ne diffèrent pas de celle trouvées dans la simulation sous simulink et ControlDesk.

**IV-2-Stratégie de commande d'un onduleur trois niveaux :**

La structure d'un onduleur à trois niveaux est montrée à la **figure(IV-6)**. Il comporte douze interrupteurs. Chacun d'eux est constitué d'un interrupteur commandable à l'amorçage et au blocage (**IGBT par exemple**) et d'une diode montée en tête bêche. Le tout forme un interrupteur bidirectionnel en courant. En plus, on a six diodes qui permettent d'avoir une tension de sortie nulle.



**Figure IV.6:** Structure de l'onduleur à trois niveaux à structure N.P.C. (à point milieu flottant [12]).

La loi de commande complémentaire des interrupteurs d'un bras d'onduleur à trois niveaux est :

$$\begin{cases} b_{11} = \overline{b_{14}} \\ b_{12} = \overline{b_{13}} \end{cases} \quad (\text{IV.3})$$

Les tensions de sortie de l'onduleur triphasé sont données par le système suivant :

$$\begin{cases} V_{am} = (b_{11}.b_{12} - b_{13}.b_{14}).U_c \\ V_{bm} = (b_{21}.b_{22} - b_{23}.b_{24}).U_c \\ V_{cm} = (b_{31}.b_{32} - b_{33}.b_{34}).U_c \end{cases} \quad (\text{IV.4})$$

Le principe de la stratégie triangulo-sinusoidale consiste à utiliser l'intersection d'une onde de référence ou modulante (qui est l'image de la tension de sortie qu'on veut obtenir) avec une ou plusieurs ondes de modulation ou porteuses, souvent triangulaires ou en dents de scie [10].

#### IV-2-1. Commande triangulo-sinusoidale à une seule porteuse:

##### IV-2-1-1. Principe de la stratégie:

Le principe de cette commande est de comparer une tension de référence avec une onde triangulaire ou en dent de scie unipolaire (**Figure IV.7**).

Pour l'onduleur triphasé à trois niveaux, les tensions de référence du système triphasé sont données par le système d'équations suivant :

$$\begin{cases} V_{ref1} = V_m \sin(\omega t - \varphi) \\ V_{ref2} = V_m \sin\left(\omega t - \varphi - \frac{2\pi}{3}\right) \\ V_{ref3} = V_m \sin\left(\omega t - \varphi - \frac{4\pi}{3}\right) \end{cases} \quad (\text{IV.5})$$

L'équation de la porteuse est donné par :

$$U_p = \begin{cases} 2U_{pm} \frac{t}{T_p} & \text{pour } 0 \leq t \leq \frac{T_p}{2} \\ 2U_{pm} \left[1 - \frac{t}{T_p}\right] & \text{pour } \frac{T_p}{2} \leq t \leq T_p \end{cases} \quad (\text{IV.6})$$

**IV-2-1-2. Algorithme de commande de la stratégie :**

Les signaux de commande de la stratégie triangulo-sinusoïdale à une seule porteuse unipolaire se déduisent de l'algorithme suivant :

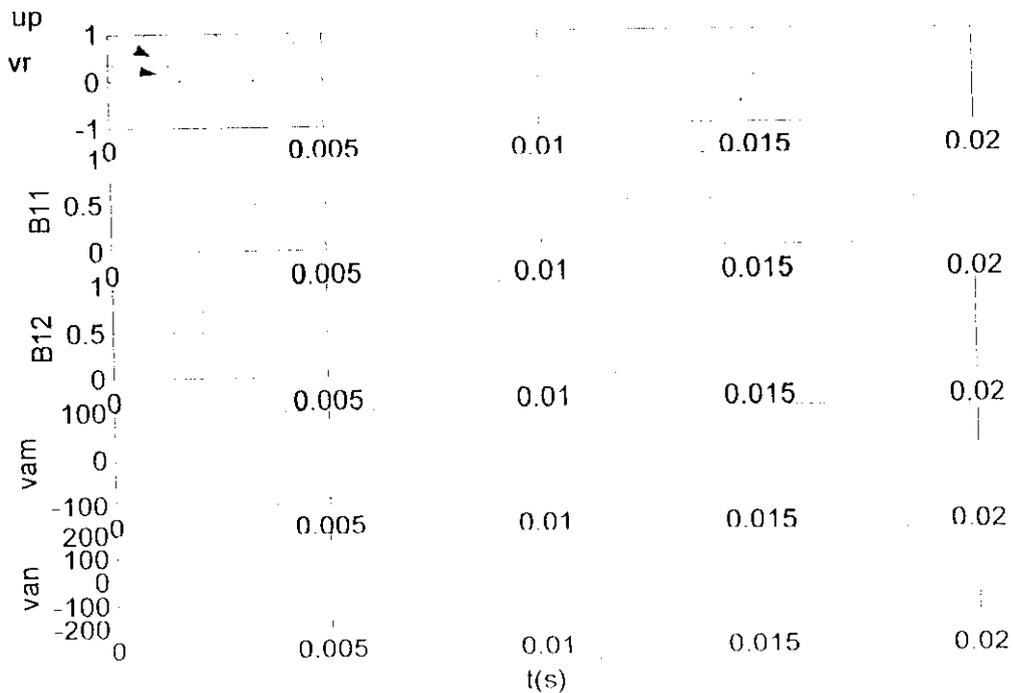
$$\begin{cases} (|V_{refk}| \leq U_p) \Rightarrow B_{K1} = 1, B_{K2} = 0 \\ (|V_{refk}| > U_p) \& (V_{refk} > 0) \Rightarrow B_{K1} = 1, B_{K2} = 1 \\ (|V_{refk}| > U_p) \& (V_{refk} < 0) \Rightarrow B_{K1} = 0, B_{K2} = 0 \end{cases} \quad (IV.7)$$

Avec cet algorithme et pour la première phase ( $\varphi = 0$ ), nous pouvons écrire :

- pour  $0 < \omega t \leq \pi \Rightarrow B_{11} = 1$ , et on module la tension de sortie en jouant sur  $B_{12}$ .
- pour  $\pi < \omega t < 2\pi \Rightarrow B_{12} = 0$ , et on module la tension de sortie en jouant sur  $B_{11}$ .

**VI-2-1-3. Résultats de simulation :**

Les signaux de commande de la stratégie triangulo-sinusoïdale à une seule porteuse unipolaire et la tension de sortie sont montrés pour  $m=12$ ,  $r=0.8$  (Fig.IV.7) et  $r=0.2$  (Fig.IV.8).



**Figure IV.7:** Signaux de commande et la tension de sortie de l'onduleur triphasé à trois niveaux commandé par la stratégie triangulo-sinusoïdale à une seule porteuse ( $m=12$ ,  $r=0.8$ ).

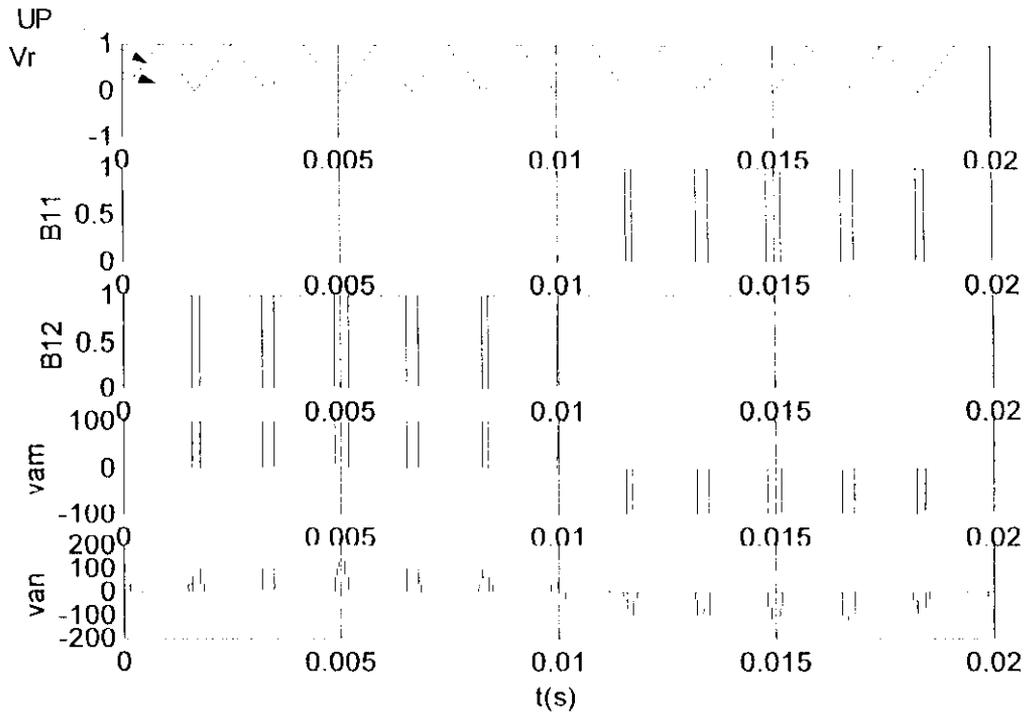


Figure IV.8: Signaux de commande et la tension de sortie de l'onduleur triphasé à trois niveaux commandé par la stratégie triangulo-sinusoïdale à une seule porteuse ( $m=12$ ,  $r=0.2$ ).

IV-2-1-4. Acquisition des signaux MLI sur la carte dSPACE :

En procédant selon l'organigramme présenté à l'annexe III, on obtient les résultats montrés ci dessous . Ces résultats sont obtenus sous ControlDesk (FigIV.9) . Les mêmes signaux sont visualisés sur l'oscilloscope (FigIV.10). Ces résultats sont obtenus en prenant  $m=12$  et  $r=0.8$  et  $r=0.2$ .

Note :On fait varier le r directement du panneaux d'instrumentation grâce au slider [7].

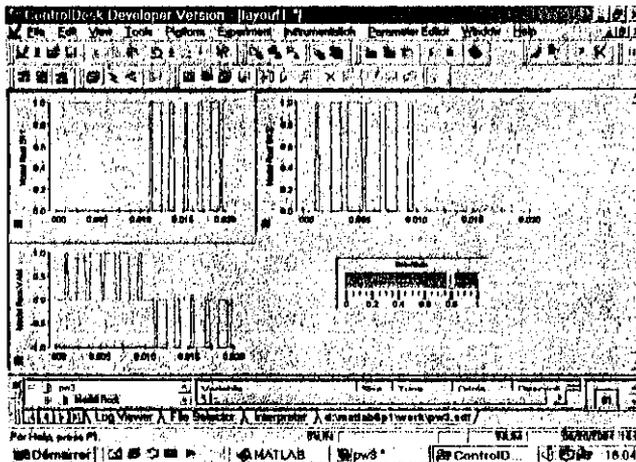


Figure IV.9.1 : MLI d'un onduleur trois niveaux à une seule porteuse ( $r=0.8$  et  $m=12$ ).

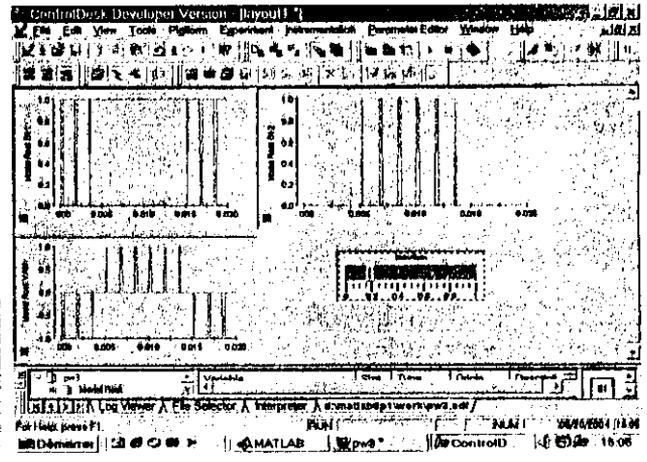
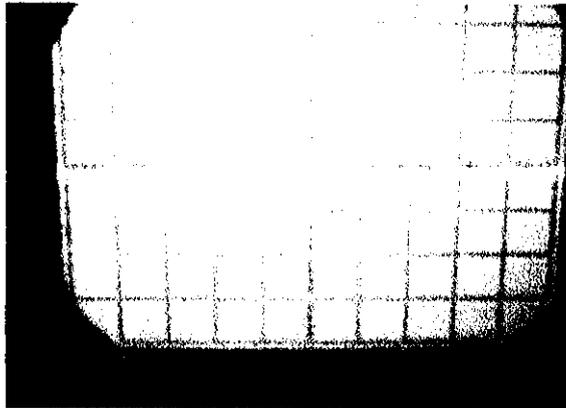


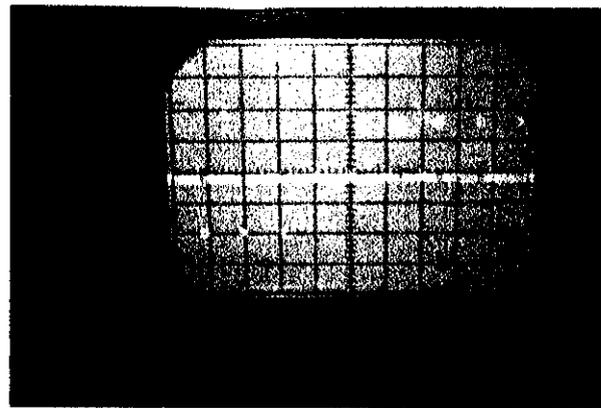
Figure IV.9.2 : MLI d'un onduleur trois niveaux à une seule porteuse ( $r=0.2$  et  $m=12$ ).

Figure IV.9

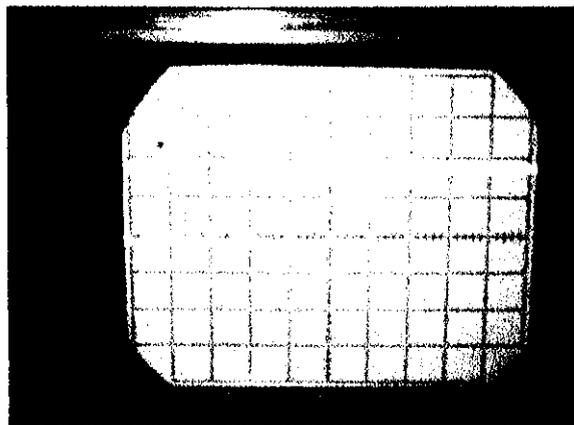
La figure suivante (Fig IV.10) représente les signaux de commande des bases des transistors de l'onduleur trois niveaux et la tension de sortie pour la stratégie triangulo-sinusoidale à une porteuse unipolaire, pour  $m=12$ ,  $r=0.8$  et  $r=0.2$ .



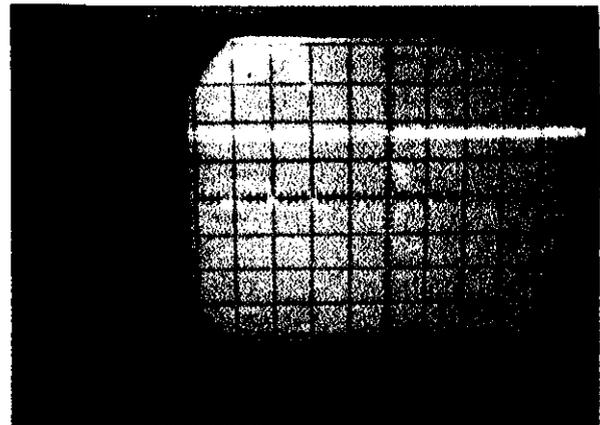
a) Forme de  $V_{am}$  pour  $m=12$  ;  $r=0,8$



b) Forme de  $V_{am}$  pour  $m=12$  ;  $r=0,2$



c) Forme de  $b_{11}$  pour  $m=12$  ;  $r=0,8$ .



e) Forme de  $b_{11}$  pour  $m=12$  ;  $r=0,8$ .

**Figure IV.10** : visualisation des résultats obtenus sur l'oscilloscope.

**Remarques :**

- En diminuant  $r$  (tous en gardant  $m=12$ ), on remarque que la largeur des signaux de commande et des tensions de sortie diminuent.
- A valeurs données de  $m$  ( $m=12$ ) et de  $r$  :

Les formes d'ondes des tensions et des signaux de commande ne diffèrent pas de celles trouvées dans la simulation sous simulink et ControlDesk.

### IV-2-2. Commande triangulo-sinusoïdale à deux porteuses bipolaires

Afin d'améliorer le taux d'harmoniques des tensions de sortie de l'onduleur à trois niveaux, on peut utiliser deux porteuses identiques décalées entre elle d'une demi-période  $T_p/2$ .

Différentes porteuses sont possibles :

- triangulaire bipolaire .
- dent de scie bipolaire .

Cette stratégie est caractérisée comme la stratégie précédente par l'indice de modulation  $m$  et le taux de modulation  $r$ .

Pour cette commande, on étudie le cas de deux porteuses en dent de scie bipolaires. Les tensions de référence de cette stratégie sont les mêmes que la stratégie précédente définie par le système (IV.5).

#### IV-2-2-1-Principe de la stratégie

Puisque l'onduleur à trois niveaux est équivalent à deux onduleurs à deux niveaux mis en série [13], l'algorithme de la stratégie pour un bras d'onduleur  $k$  peut être résumé en deux étapes:

**Etape 1 :** Détermination des signaux de commande intermédiaires  $V_{K0}, V_{K1}$  :

$$\begin{cases} V_{refk} \geq U_{p1} \Rightarrow V_{K1} = U_C \\ V_{refk} < U_{p1} \Rightarrow V_{K1} = 0 \end{cases} \quad (IV.8)$$

$$\begin{cases} V_{refk} \geq U_{p2} \Rightarrow V_{K0} = 0 \\ V_{refk} < U_{p2} \Rightarrow V_{K0} = -U_C \end{cases} \quad (IV.9)$$

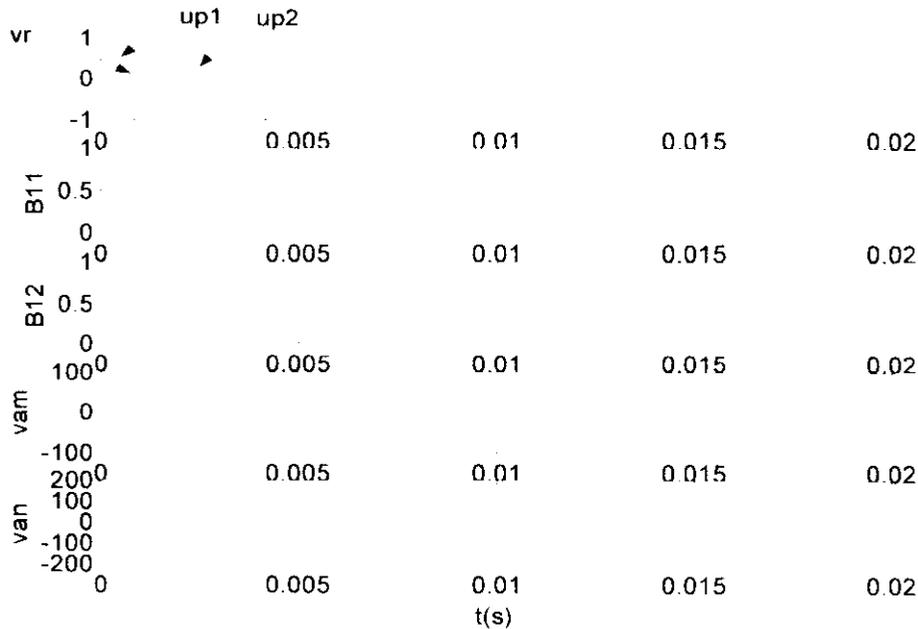
**Etape 2 :** Détermination du signal  $V_{Km}$  et les ordres de commande  $B_{Ks}$  des interrupteurs :

$$\begin{cases} V_{Km} = U_C \Rightarrow B_{K1} = 1, B_{K2} = 1 \\ V_{Km} = -U_C \Rightarrow B_{K1} = 0, B_{K2} = 0 \\ V_{Km} = 0 \Rightarrow B_{K1} = 1, B_{K2} = 0 \end{cases} \quad (IV.10)$$

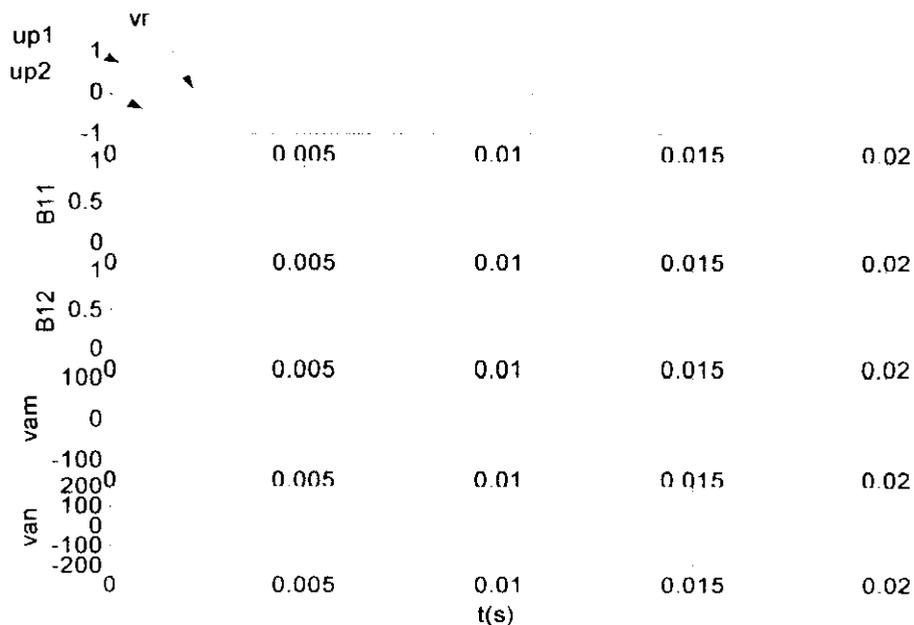
$$\text{Avec } \begin{cases} V_{Km} = V_{K1} + V_{K0} \\ B_{K1} = B_{K1} \\ B_{K2} = \overline{B_{K1}} \end{cases} \quad (IV.11)$$

**IV-2-2-2-Simulation**

Les figures (IV.11) et (IV.12) montrent les signaux de commande et la tension de sortie de l'onduleur à trois niveaux commandé par la stratégie triangulo-sinusoïdale à deux porteuses bipolaires pour  $m=12$ , et  $r=0.8$  et  $r=0.2$  respectivement.



**Figure IV.11:** Signaux de commande et tension de sortie de l'onduleur triphasé à trois niveaux commandé par la stratégie triangulo-sinusoïdale à deux porteuses bipolaires ( $m=12, r=0.8$ )



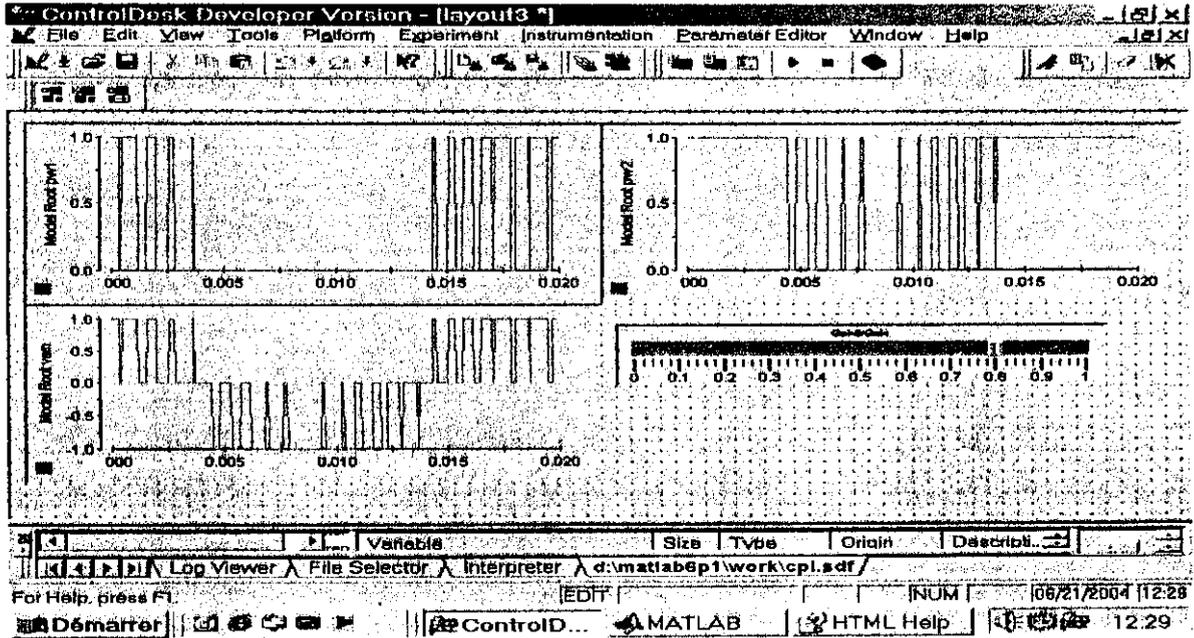
**Figure IV.12:** Signaux de commande et tension de sortie de l'onduleur triphasé à trois niveaux commandé par la stratégie triangulo-sinusoïdale à deux porteuses bipolaires ( $m=12, r=0.2$ ).

**IV-2-2-3-Acquisition des signaux MLI sur la carte dSPACE:**

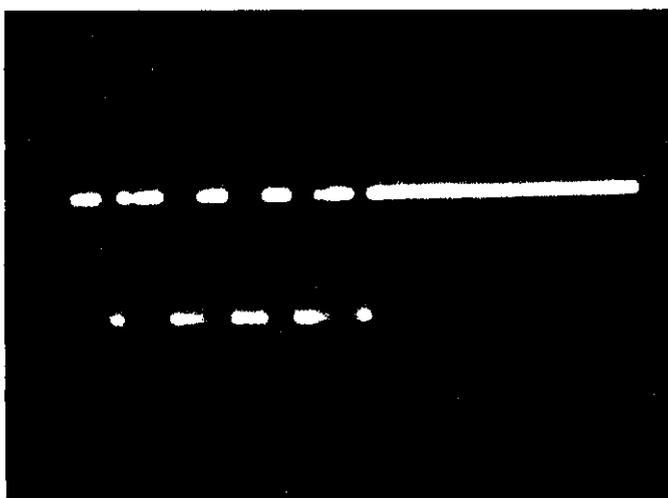
En procédant selon l'organigramme présenté à l'annexe III, on obtient les résultats montrés ci dessous. Ces résultats sont obtenus sous ControlDesk (Fig.IV.13).

Les mêmes signaux sont visualisés sur l'oscilloscope (Fig.IV.14).

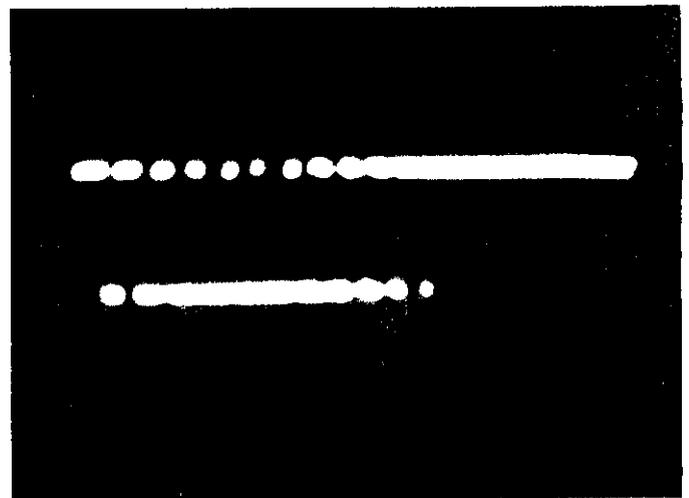
Ces résultats sont obtenus en prenant  $m=12$  et  $r=0.8$ .



**Figure IV.13 :** MLI triangulo-sinusoidale à deux porteuses bipolaires ( $m=12$  et  $r=0.8$ ).



**a)** Forme du signal **b12** ( $m=12$  et  $r=0.8$ ).



**b)** Forme du signal **b11** ( $m=12$  et  $r=0.8$ ).

**Figure IV.14 :** visualisation des résultats obtenus sur l'oscilloscope.

**IV-3. Interprétation des résultats :**

On constate pour cette stratégie que :

- L'augmentation du rapport cyclique  $r$  permet de diminuer le taux d'harmoniques de la tension de sortie.

**Conclusion:**

Dans ce chapitre, On a étudié et détaillé l'implémentation d'algorithmes de commande des onduleurs deux et trois niveaux sur la carte dSPACE DS1104 pour la génération des signaux de commande MLI.

Les résultats obtenus dans les essais pratiques montrent que la stratégie triangulo-sinusoidale utilisant une ou deux porteuses peut être effectuée en numérique (avec échantillonnage régulier des tensions de références) et donne de bons résultats.

Cette partie nous a fait sentir l'importance d'une telle carte dans l'élaboration de nouvelles stratégies de commande des systèmes.

## Conclusion générale

L'objectif principal du travail présenté dans ce mémoire est la mise en œuvre de la carte dSPACE DS1104 ainsi que l'implémentation d'algorithmes de commande générant une MLI en temps réel pour les onduleurs deux et trois niveaux.

Afin d'aborder ce travail, nous avons commencé par une description générale des DSPs. Puis, on a spécifié les différentes caractéristiques de notre carte en détaillant le fonctionnement de ces deux processeurs (Master MPC8240 et Slave TMS320F240).

Ensuite, on a décrit l'environnement logiciel en se basant sur la programmation via simulink et on précisant les différentes étapes de compilation et d'exécution.

Dans le dernier chapitre, on a utilisé la carte pour la commande rapprochée des onduleurs à deux et trois niveaux. Pour se faire, on a implémenté les deux stratégies suivantes :

- Triangulo-sinusoidale à une porteuse, pour les onduleurs deux et trois niveaux.
- Triangulo-sinusoidale à deux porteuses, pour les onduleurs trois niveaux.

Cette partie nous a permis d'utiliser la carte et de juger son importance dans la commande de ce type de système.

En perspective de ce travail on propose les idées suivantes :

- Implémentation d'autres algorithmes MLI.
- Utilisation du langage assembleur sans passage par simulink.

## Bibliographie

- [1] : G.Baudoin et F.Viroleau :  
«Les processeurs de traitement du signal TMS320C5X », pp.15-220, CCU.
- [2] : Henri LILEN : « Microprocesseurs », pp.363-366, CCF.
- [3] : Guy SEGUIER et Francis LABRIQUE « Les convertisseurs de l'électronique de puissance », pp.10-60, bibliothèque ENP.
- [4] : Le livre sous la cote : 629.05 STU « Modélisation et commande des moteurs triphasés », pp.160-208, bibliothèque ENP.
- [5] : E.M.Berkouk : « Contribution à la conduite des machines asynchrones monophasées et triphasées alimentées par des convertisseurs directs et indirects », Thèse de Doctorat, Paris.
- [6] : Photo DS1104 R&D fournie par dSPACE.
- [7] : Experiment guide (Manuel de la carte dSPACE).
- [8] : Installation and configuration guide (Manuel de la carte dSPACE).
- [9] : TMS320F240 DSP CONTROLLER :  
« SPRS042E-OCTOBER 1996-Revised number2002 ».
- [10] : M.Bilaid, N.Bouaicha, F.Rahou : « Réalisation de la commande triangulo-sinusoidale à une porteuse pour un onduleur à trois niveaux », Mémoire de DEUA,USTHB 2002.
- [11] : R.Kerid, M.Gheraba, « Commande de vitesse et de position par réseau de neurones d'une MAS », Thèse de magister, pp. 2-60.
- [12] : M.Gaad « Modélisation et réalisation d'un onduleur triphasé à trois niveaux à structure NPC. Application à la conduite de la machine asynchrone », Thèse de magister, EMP, 2000.
- [13] : MATLAB-Dspace « Interface Libraries ».
- [14] : Implementation guide (Manuel de la carte dSPACE).
- [15] : Fichier PDF : DS1104RTI (documentation de dSPACE).
- [16] : [www.dspace.de/solutions\\_of\\_control](http://www.dspace.de/solutions_of_control).
- [17] : [www.motorola.com](http://www.motorola.com) : MPC8240 Integrated Processor Hardware Specification (MPC8240EC.PDF).

# Annexes

Le jeu d'instruction du TMS320F240, avec la description et le code de chaque instruction sont spécifiés dans les tableaux suivants :

F240 MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
ABS	Absolute value of accumulator	1/1	1011	1110	0000	0000
ADD	Add to accumulator with shift	1/1	0010	SHFT	IADD	RESS
	Add to high accumulator	1/1	0110	0001	IADD	RESS
	Add to accumulator short immediate	1/1	1011	1000	KKKK	KKKK
	Add to accumulator long immediate with shift	2/2	1011	1111	1001	SHFT
ADDC	Add to accumulator with carry	1/1	0110	0000	IADD	RESS
ADDS	Add to low accumulator with sign extension suppressed	1/1	0110	0010	IADD	RESS
ADDT	Add to accumulator with shift specified by T register	1/1	0110	0011	IADD	RESS
ADRK	Add to auxiliary register short immediate	1/1	0111	1000	KKKK	KKKK
AND	AND with accumulator	1/1	0110	1110	IADD	RESS
	AND immediate with accumulator with shift	2/2	1011	1111	1011	SHFT 16-Bit Constant
	AND immediate with accumulator with shift of 16	2/2	1011	1110	1000	0001 16-Bit Constant
APAC	Add P register to accumulator	1/1	1011	1110	0000	0100
B	Branch unconditionally	2/4	0111	1001	IADD	RESS Branch Address
BACC	Branch to address specified by accumulator	1/4	1011	1110	0010	0000
BANZ	Branch on auxiliary register not zero	2/4/2	0111	1011	IADD	RESS Branch Address
BCND	Branch if TC bit $\neq$ 0	2/4/2	1110	0001	0000	0000 Branch Address
	Branch if TC bit = 0	2/4/2	1110	0010	0000	0000 Branch Address
	Branch on carry	2/4/2	1110	0011	0001	0001 Branch Address
	Branch if accumulator $\geq$ 0	2/4/2	1110	0011	1000	1100 Branch Address
	Branch if accumulator > 0	2/4/2	1110	0011	0000	0100 Branch Address
	Branch on I/O status low	2/4/3	1110	0000	0000	0000 Branch Address
	Branch if accumulator $\leq$ 0	2/4/2	1110	0011	1100	1100 Branch Address
	Branch if accumulator < 0	2/4/2	1110	0011	0100	0100 Branch Address
	Branch on no carry	2/4/2	1110	0011	0000	0001 Branch Address
	Branch if no overflow	2/4/2	1110	0011	0000	0010 Branch Address



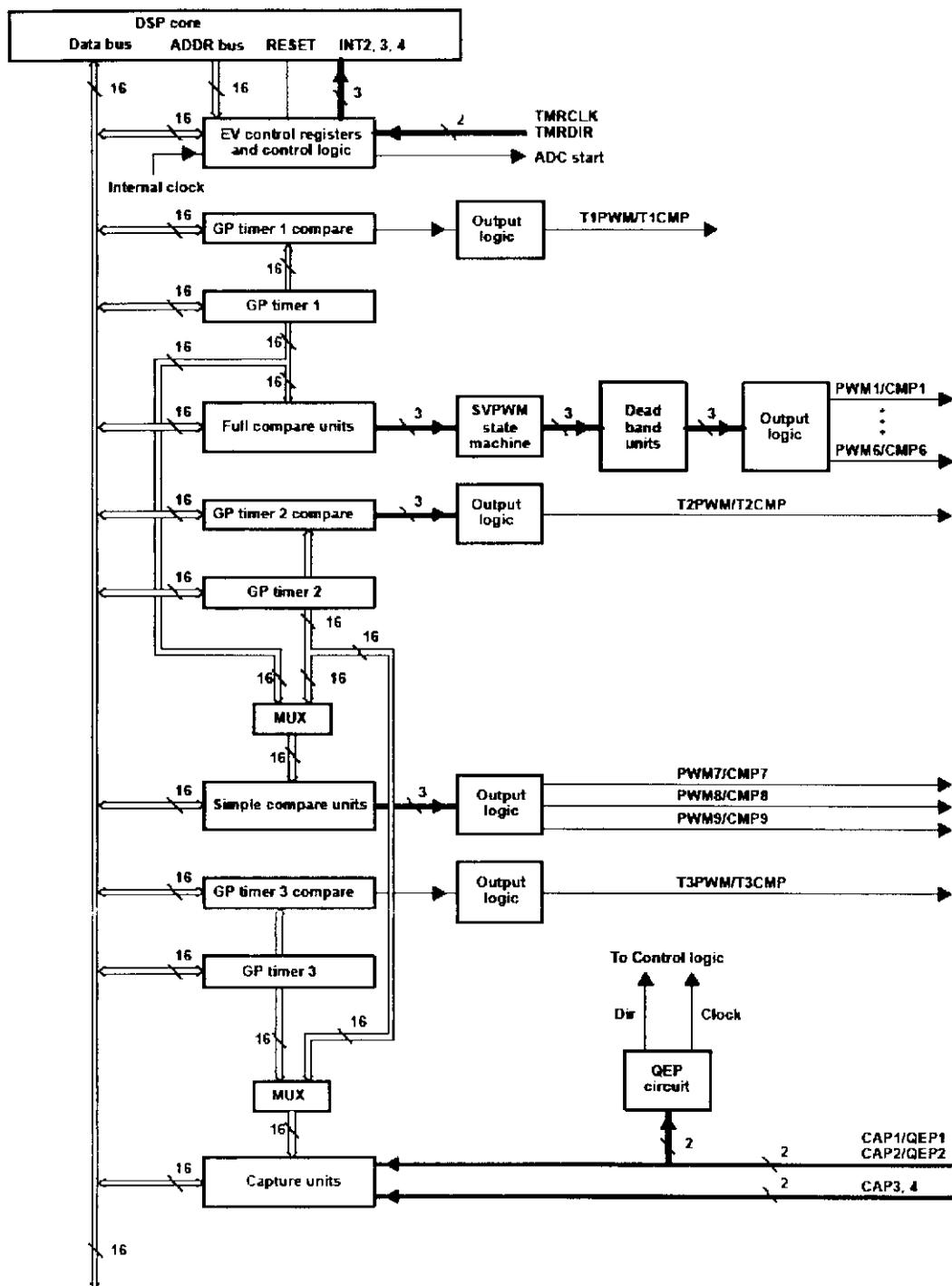
F240 MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
BCND	Branch if accumulator $\neq$ 0	2/4/2	1110	0011	0000	1000
			Branch Address			
	Branch on overflow	2/4/2	1110	0011	0010	0010
			Branch Address			
	Branch if accumulator = 0	2/4/2	1110	0011	1000	1000
			Branch Address			
BIT	Test bit	1/1	0100	BITx	IADD	RESS
BITT	Test bit specified by TREG	1/1	0110	1111	IADD	RESS
BLDD†	Block move from data memory to data memory source immediate	2/3	1010	1000	IADD	RESS
			Branch Address			
	Block move from data memory to data memory destination immediate	2/3	1010	1001	IADD	RESS
			Branch Address			
BLPD	Block move from program memory to data memory	2/3	1010	0101	IADD	RESS
			Branch Address			
CALA	Call subroutine indirect	1/4	1011	1110	0011	0000
CALL	Call subroutine	2/4	0111	1010	IADD	RESS
			Routine Address			
CC	Conditional call subroutine	2/4/2	1110	10TP	ZLVC	ZLVC
			Routine Address			
CLRC	Configure block as data memory	1/1	1011	1110	0100	0100
	Enable interrupt	1/1	1011	1110	0100	0000
	Reset carry bit	1/1	1011	1110	0100	1110
	Reset overflow mode	1/1	1011	1110	0100	0010
	Reset sign-extension mode	1/1	1011	1110	0100	0110
	Reset test/control flag	1/1	1011	1110	0100	1010
	Reset external flag	1/1	1011	1110	0100	1100
CMPPL	Complement accumulator	1/1	1011	1110	0000	0001
CMPR	Compare auxiliary register with auxiliary register AR0	1/1	1011	1111	0100	01CM
DMOV	Data move in data memory	1/1	0111	0111	IADD	RESS
IDLE	Idle until interrupt	1/1	1011	1110	0010	0010
IN	Input data from port	2/2	1010	1111	IADD	RESS
			16BIT	I/O	PORT	ADRS
INTR	Software-interrupt	1/4	1011	1110	011K	KKKK
LACC	Load accumulator with shift	1/1	0001	SHFT	IADD	RESS
	Load accumulator long immediate with shift	2/2	1011	1111	1000	SHFT
			16-Bit Constant			
	Zero low accumulator and load high accumulator	1/1	0110	1010	IADD	RESS

F240 MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
LACL	Load accumulator immediate short	1/1	1011	1001	KKKK	KKKK
	Zero accumulator	1/1	1011	1001	0000	0000
	Zero low accumulator and load high accumulator	1/1	0110	1010	IADD	RESS
	Zero low accumulator and load low accumulator with no sign extension	1/1	0110	1001	IADD	RESS
LACT	Load accumulator with shift specified by T register	1/1	0110	1011	IADD	RESS
LAR	Load auxiliary register	1/2	0000	0ARx	IADD	RESS
	Load auxiliary register short immediate	1/2	1011	0ARx	KKKK	KKKK
	Load auxiliary register long immediate	2/2	1011	1111	0000	1ARx 16-Bit Constant
LDP	Load data-memory page pointer	1/2	0000	1101	IADD	RESS
	Load data-memory page pointer immediate	1/2	1011	110P	AGEP	OINT
LPH	Load high-P register	1/1	0111	0101	IADD	RESS
LST	Load status register ST0	1/2	0000	1110	IADD	RESS
	Load status register ST1	1/2	0000	1111	IADD	RESS
LT	Load TREG	1/1	0111	0011	IADD	RESS
LTA	Load TREG and accumulate previous product	1/1	0111	0000	IADD	RESS
LTD	Load TREG, accumulate previous product, and move data	1/1	0111	0010	IADD	RESS
LTP	Load TREG and store P register in accumulator	1/1	0111	0001	IADD	RESS
LTS	Load TREG and subtract previous product	1/1	0111	0100	IADD	RESS
MAC	Multiply and accumulate	2/3	1010	0010	IADD	RESS 16-Bit Constant
MACD	Multiply and accumulate with data move	2/3	1010	0011	IADD	RESS 16-Bit Constant
MAR	Load auxiliary register pointer	1/1	1000	1011	1000	1ARx
	Modify auxiliary register	1/1	1000	1011	IADD	RESS
MPY	Multiply (with TREG, store product in P register)	1/1	0101	0100	IADD	RESS
	Multiply immediate	1/1	110C	KKKK	KKKK	KKKK
MPYA	Multiply and accumulate previous product	1/1	0101	0000	IADD	RESS
MPYS	Multiply and subtract previous product	1/1	0101	0001	IADD	RESS
MPYU	Multiply unsigned	1/1	0101	0101	IADD	RESS
NEG	Negate accumulator	1/1	1011	1110	0000	0010
NMI	Nonmaskable interrupt	1/4	1011	1110	0101	0010
NOP	No operation	1/1	1000	1011	0000	0000
NORM	Normalize contents of accumulator	1/1	1010	0000	IADD	RESS
OR	OR with accumulator	1/1	0110	1101	IADD	RESS
	OR immediate with accumulator with shift	2/2	1011	1111	1100	SHIFT 16-Bit Constant
	OR immediate with accumulator with shift of 16	2/2	1011	1110	1000	0010 16-Bit Constant
OUT	Output data to port	2/3	0000	1100	IADD	RESS 16BIT I/O PORT ADRS
PAC	Load accumulator with P register	1/1	1011	1110	0000	0011

F240 MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB		LSB	
POP	Pop top of stack to low accumulator	1/1	1011	1110	0011	0010
POPD	Pop top of stack to data memory	1/1	1000	1010	IADD	RESS
PSHD	Push data-memory value onto stack	1/1	0111	0110	IADD	RESS
PUSH	Push low accumulator onto stack	1/1	1011	1110	0011	1100
RET	Return from subroutine	1/4	1110	1111	0000	0000
RETC	Conditional return from subroutine	1/4/2	1110	11TP	ZLVC	ZLVC
ROL	Rotate accumulator left	1/1	1011	1110	0000	1100
ROR	Rotate accumulator right	1/1	1011	1110	0000	1101
RPT	Repeat instruction as specified by data-memory value	1/1	0000	1011	IADD	RESS
	Repeat instruction as specified by immediate value	1/1	1011	1011	KKKK	KKKK
SACH	Store high accumulator with shift	1/1	1001	1SHF	IADD	RESS
SACL	Store low accumulator with shift	1/1	1001	0SHF	IADD	RESS
SAR	Store auxiliary register	1/1	1000	0ARx	IADD	RESS
SBRK	Subtract from auxiliary register short immediate	1/1	0111	1100	KKKK	KKKK
SETC	Set carry bit	1/1	1011	1110	0100	1111
	Configure block as program memory	1/1	1011	1110	0100	0101
	Disable interrupt	1/1	1011	1110	0100	0001
	Set overflow mode	1/1	1011	1110	0100	0011
	Set test/control flag	1/1	1011	1110	0100	1011
	Set external flag XF	1/1	1011	1110	0100	1101
	Set sign-extension mode	1/1	1011	1110	0100	0111
SFL	Shift accumulator left	1/1	1011	1110	0000	1001
SFR	Shift accumulator right	1/1	1011	1110	0000	1010
SPAC	Subtract P register from accumulator	1/1	1011	1110	0000	0101
SPH	Store high-P register	1/1	1000	1101	IADD	RESS
SPL	Store low-P register	1/1	1000	1100	IADD	RESS
SPM	Set P register output shift mode	1/1	1011	1111	IADD	RESS
SQRA	Square and accumulate	1/1	0101	0010	IADD	RESS
SQRS	Square and subtract previous product from accumulator	1/1	0101	0011	IADD	RESS
SST	Store status register ST0	1/1	1000	1110	IADD	RESS
	Store status register ST1	1/1	1000	1111	IADD	RESS
SPLK	Store long immediate to data memory	2/2	1010	1110	IADD	RESS
			16-Bit Constant			
SUB	Subtract from accumulator long immediate with shift	2/2	1011	1111	1010	SHFT
			16-Bit Constant			
	Subtract from accumulator with shift	1/1	0011	SHFT	IADD	RESS
	Subtract from high accumulator	1/1	0110	0101	IADD	RESS
	Subtract from accumulator short immediate	1/1	1011	1010	KKKK	KKKK

F240 MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB		LSB	
SUBB	Subtract from accumulator with borrow	1/1	0110	0100	IADD	RESS
SUBC	Conditional subtract	1/1	0000	1010	IADD	RESS
SUBS	Subtract from low accumulator with sign extension suppressed	1/1	0110	0110	IADD	RESS
SUBT	Subtract from accumulator with shift specified by TREG	1/1	0110	0111	IADD	RESS
TBLR	Table read	1/3	1010	0110	IADD	RESS
TBLW	Table write	1/3	1010	0111	IADD	RESS
TRAP	Software interrupt	1/4	1011	1110	0101	0001
XOR	Exclusive-OR with accumulator	1/1	0110	1100	IADD	RESS
	Exclusive-OR immediate with accumulator with shift	2/2	1011	1111	1101	SHFT
	Exclusive-OR immediate with accumulator with shift of 16	2/2	1011	1110	1000	0011
ZALR	Zero low accumulator and load high accumulator with rounding	1/1	0110	1000	IADD	RESS

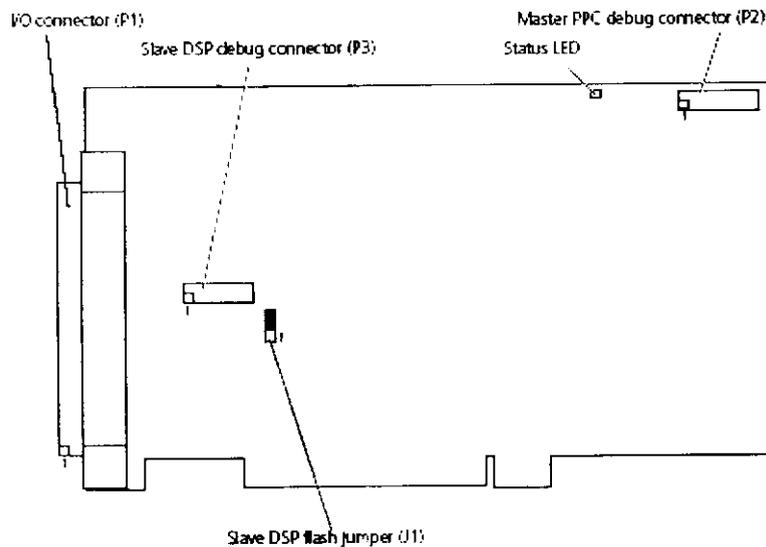




Gestionnaire d'évènements



## Connector Pinouts DS1104



The illustration shows the locations of the connectors on the DS1104. These are the most important connectors and jumpers:

- The **I/O connector (P1)** is a 100-pin connector. The signals available at this connector are described in *I/O Connector (P1)* on page 127.
- The **Master PPC debug connector (P2)** is a JTAG 1149.1 hardware debug connector and can be used for master PPC debugging. The signals available at this connector are described in *Master PPC Debug Connector (P2)* on page 132.
- The **Slave DSP debug connector (P3)** is a JTAG 1149.1 hardware debug connector and can be used for slave DSP debugging. The signals available at this connector are described in *Slave DSP Debug Connector (P3)* on page 133.



- The **Slave DSP flash jumper (J1)** is used to control the protection mode of the flash memory and the watchdog mode: see *Slave DSP Flash Jumper (J1)* on page 134.



The DS1104 is equipped with a red Status LED for troubleshooting purposes. For details, see *Checking the DS1104* on page 164.

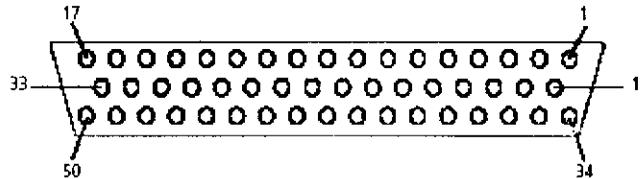
## I/O Connector (P1)

The table below shows the specifications of the signals sorted by the pin numbers of the board's 100-pin I/O connector P1. For the location of the connector on the board, see *Connector Pinouts DS1104* on page 126.

Sub-D connectors	Using the adapter cables supplied with the board, the P1 I/O connector can be linked to two 50-pin, female Sub-D connectors (labeled P1A, P1B). The table below also provides the corresponding mappings.
Pin numbering of Sub-D connectors	Because the pin numbering used for Sub-D connectors is not standardized, the following figure shows the numbering used (viewed from the top of a female connector).



**Do not rely on the numbers written on Sub-D connectors.**





Due to the board's limited number of I/O pins, some functions use the same pins. For details, see *Shared I/O Pins* on page 152.

PI 1	PIB 1	See below	GND		
PI 2	PIA 1	See below	GND		
PI 3	PIB 34	CF21 1 and CF22 8	DCD (CTS)	RS232 data carrier detect (RS422/RS485 clear to send)	RS232 (RS422/RS485)
PI 4	PIA 34	CF22 7	(RTS)	RS422/RS485 request to send (not used for RS232)	RS422/RS485
PI 5	PIB 18	CF21 8 and CF22 9	CTS (VCTS)	RS232/RS422/RS485 clear to send	RS232 (RS422/RS485)
PI 6	PIA 18	CF21 7 and CF22 6	RTS (VRTS)	RS232/RS422/RS485 request to send	RS232 (RS422/RS485)
PI 7	PIB 2	CF21 6 and CF22 3	DSR (RXD)	RS232 data set ready (RS422/RS485 receive)	RS232 (RS422/RS485)
PI 8	PIA 2	CF21 4 and CF22 2	DTR (TXD)	RS232 data terminal ready (RS422/RS485 transmit)	RS232 (RS422/RS485)
PI 9	PIB 35	CF21 2 and CF22 4	RXD (RXD)	RS232/RS422/RS485 receive	RS232 (RS422/RS485)
PI 10	PIA 35	CF21 3 and CF22 1	TXD (TXD)	RS232/RS422/RS485 transmit	RS232 (RS422/RS485)
PI 11	PIB 19	CP18 34	SSOM	SPI slave out, master in (bit I/O)	TTL±13 mA (TTL±13 mA)
PI 12	PIA 19	CP18 22	SCAP4	Capture input (bit I/O)	TTL/no (TTL±13 mA)
PI 13	PIB 3	CP18 16	SSIMO	SPI slave in, master out (bit I/O)	TTL±13 mA (TTL±13 mA)
PI 14	PIA 3	CP18 3	SCAP3	Capture input (bit I/O)	TTL/no (TTL±13 mA)
PI 15	PIB 36	CP18 35	SSTE	SPI slave transmit enable (bit I/O)	TTL±13 mA (TTL±13 mA)
PI 16	PIA 36	CP18 21	SCAP2	Capture input (bit I/O)	TTL/no (TTL±13 mA)
PI 17	PIB 20	CP18 17	SCLK	SPI clock (bit I/O)	TTL±13 mA (TTL±13 mA)
PI 18	PIA 20	CP18 2	SCAP1	Capture input (bit I/O)	TTL/no (TTL±13 mA)
PI 19	PIB 4	See below	VCC	PC 5 V power supply	5 V ±10% / total load of all pins providing VCC: max. 500 mA
PI 20	PIA 4	See below	VCC	PC 5 V power supply	5 V ±10% / total load of all pins providing VCC: max. 500 mA
PI 21	PIB 37	CP18 24	ST3PAM	Bit I/O	TTL±13 mA
PI 22	PIA 37	CP18 28	SPWM6	PWM output	TTL±13 mA
PI 23	PIB 21	CP18 5	ST2PAM	PWM output (bit I/O or simple compare)	TTL±13 mA (TTL±13 mA or no)
PI 24	PIA 21	CP18 9	SPWM5	PWM output (simple compare)	TTL±13 mA (TTL/no)
PI 25	PIB 5	CP18 23	ST1PAM	PWM interrupt output (external interrupt input or bit I/O)	TTL±13 mA (TTL/no)



I/O Pins			Description / Function of Pin		
P1 26	P1A 5	CP18 27	SPWM4	PWM output	TTL/±13 mA
P1 27	P1B 38	CP18 11	SPWM9	PWM output (bit I/O)	TTL/±13 mA (TTL/±13 mA)
P1 28	P1A 38	CP18 8	SPWM3	PWM output (simple compare)	TTL/±13 mA (TTL/no)
P1 29	P1B 22	CP18 29	SPWM8	PWM output (bit I/O)	TTL/±13 mA (TTL/±13 mA)
P1 30	P1A 22	CP18 26	SPWM2	PWM output	TTL/±13 mA
P1 31	P1B 6	CP18 10	SPWM7	PWM output (bit I/O)	TTL/±13 mA (TTL/±13 mA)
P1 32	P1A 6	CP18 7	SPWM1	PWM output (simple compare)	TTL/±13 mA (TTL/no)
P1 33	P1B 39	See below	GND		
P1 34	P1A 39	See below	GND		
P1 35	P1B 23	CP20 7	IDX(2)	Digital incremental enc. interface input index (inverted signal)	TTL or RS422
P1 36	P1A 23	CP19 7	IDX(1)	Digital incremental enc. interface input index (inverted signal)	TTL or RS422
P1 37	P1B 7	CP20 6	IDX(2)	Digital incremental enc. interface input index	TTL or RS422
P1 38	P1A 7	CP19 6	IDX(1)	Digital incremental enc. interface input index	TTL or RS422
P1 39	P1B 40	CP20 5	/PHB(2)	Digital incremental enc. interface input 90° (inverted signal)	TTL or RS422
P1 40	P1A 40	CP19 5	/PHB(1)	Digital incremental enc. interface input 90° (inverted signal)	TTL or RS422
P1 41	P1B 24	CP20 4	PHB(2)	Digital incremental enc. interface input 90°	TTL or RS422
P1 42	P1A 24	CP19 4	PHB(1)	Digital incremental enc. interface input 90°	TTL or RS422
P1 43	P1B 8	CP20 3	/PHD(2)	Digital incremental enc. interface input 0° (inverted signal)	TTL or RS422
P1 44	P1A 8	CP19 3	/PHD(1)	Digital incremental enc. interface input 0° (inverted signal)	TTL or RS422
P1 45	P1B 41	CP20 2	PHD(2)	Digital incremental enc. interface input 0°	TTL or RS422
P1 46	P1A 41	CP19 2	PHD(1)	Digital incremental enc. interface input 0°	TTL or RS422
P1 47	P1B 25	See below	GND		
P1 48	P1A 25	See below	GND		
P1 49	P1B 9	CP17 15	ID19	Bit I/O (external interrupt input)	TTL/±5 mA (TTL/no)
P1 50	P1A 9	CP17 33	ID18	Bit I/O (external interrupt input)	TTL/±5 mA (TTL/no)
P1 51	P1B 42	CP17 14	ID17	Bit I/O (external interrupt input)	TTL/±5 mA (TTL/no)
P1 52	P1A 42	CP17 32	ID16	Bit I/O (external interrupt input)	TTL/±5 mA (TTL/no)
P1 53	P1B 26	CP17 12	ID15	Bit I/O	TTL/±5 mA
P1 54	P1A 26	CP17 30	ID14	Bit I/O	TTL/±5 mA
P1 55	P1B 10	CP17 11	ID13	Bit I/O	TTL/±5 mA
P1 56	P1A 10	CP17 29	ID12	Bit I/O	TTL/±5 mA
P1 57	P1B 43	CP17 9	ID11	Bit I/O	TTL/±5 mA
P1 58	P1A 43	CP17 27	ID10	Bit I/O	TTL/±5 mA
P1 59	P1B 27	CP17 8	ID9	Bit I/O	TTL/±5 mA
P1 60	P1A 27	CP17 26	ID8	Bit I/O	TTL/±5 mA
P1 61	P1B 11	CP17 6	ID7	Bit I/O	TTL/±5 mA
P1 62	P1A 11	CP17 24	ID6	Bit I/O	TTL/±5 mA



Pin	Port	Compteur	Fonction	Signal	Caractéristiques
P1 63	P1B 44	CP17 5	ID5	Bit I/O	TTL±5 mA
P1 64	P1A 44	CP17 23	ID4	Bit I/O	TTL±5 mA
P1 65	P1B 28	CP17 3	ID3	Bit I/O	TTL±5 mA
P1 66	P1A 28	CP17 21	ID2	Bit I/O	TTL±5 mA
P1 67	P1B 12	CP17 2	ID1	Bit I/O	TTL±5 mA
P1 68	P1A 12	CP17 20	ID0	Bit I/O	TTL±5 mA
P1 69	P1B 45	See below	GND		
P1 70	P1A 45	See below	GND		
P1 71	P1B 29	CP16	DACH8	16-bit DAC/10 µs	±10 V±5 mA
P1 72	P1A 29	CP15	DACH7	16-bit DAC/10 µs	±10 V±5 mA
P1 73	P1B 13	See below	GND		
P1 74	P1A 13	See below	GND		
P1 75	P1B 46	CP14	DACH5	16-bit DAC/10 µs	±10 V±5 mA
P1 76	P1A 46	CP13	DACH5	16-bit DAC/10 µs	±10 V±5 mA
P1 77	P1B 30	See below	GND		
P1 78	P1A 30	See below	GND		
P1 79	P1B 14	CP12	DACH4	16-bit DAC/10 µs	±10 V±5 mA
P1 80	P1A 14	CP11	DACH3	16-bit DAC/10 µs	±10 V±5 mA
P1 81	P1B 47	See below	GND		
P1 82	P1A 47	See below	GND		
P1 83	P1B 31	CP10	DACH2	16-bit DAC/10 µs	±10 V±5 mA
P1 84	P1A 31	CP9	DACH1	16-bit DAC/10 µs	±10 V±5 mA
P1 85	P1B 15	See below	GND		
P1 86	P1A 15	See below	GND		
P1 87	P1B 48	CP8	ADCH8	12-bit ADC/0.8 µs	±10 V
P1 88	P1A 48	CP7	ADCH7	12-bit ADC/0.8 µs	±10 V
P1 89	P1B 32	See below	GND		
P1 90	P1A 32	See below	GND		
P1 91	P1B 16	CP6	ADCH5	12-bit ADC/0.8 µs	±10 V
P1 92	P1A 16	CP5	ADCH5	12-bit ADC/0.8 µs	±10 V
P1 93	P1B 49	See below	GND		
P1 94	P1A 49	See below	GND		
P1 95	P1B 33	CP4	ADCH4	16-bit ADC/2.0 µs	±10 V
P1 96	P1A 33	CP3	ADCH3	16-bit ADC/2.0 µs	±10 V
P1 97	P1B 17	See below	GND		
P1 98	P1A 17	See below	GND		
P1 99	P1B 50	CP2	ADCH2	16-bit ADC/2.0 µs	±10 V
P1 100	P1A 50	CP1	ADCH1	16-bit ADC/2.0 µs	±10 V

Signal names in parentheses apply when the UART is set to the RS485 or RS422 mode.



