

## 1. Introduction

L'objet de cette partie est d'étudier le DSP TMS320C6701. Néanmoins les informations fournies ne sont pas exhaustives et, pour une utilisation précise de telle ou telle partie, on pourra avantageusement utiliser les documentations fournies par le constructeur.

## 2. Description générale

La plateforme TMS320C6000 des processeurs de signaux numériques fait partie de la famille TMS320. Elle comporte les processeurs TMS320C62x à arithmétique fixe et TMS320C67x à arithmétique flottante. Le TMS320C6701 est un membre de la catégorie C67x, son architecture VLIW lui permet le traitement par paquets de 8 instructions en parallèle par 8 unités fonctionnelles. Le CPU est organisée en deux blocs d'unités fonctionnelles. Le premier contient les unités fonctionnelles .L1, .S1, .M1, .D1 et 16 registres constituant le chemin A (pathA). Le deuxième contient quatre autres unités .L2, .S2, .M2, .D2 et 16 registres constituant ainsi le chemin B (path B).

## 3. Caractéristiques principales du TMS320C6701

Le DSP TMS320C6701 comportant 2 CPUs, où chacune dispose des caractéristiques suivantes :

- 16 Registres de 32 bits ;
- 04 UAL et multiplieurs ;
- 334 MIPS ;
- 600 MFLOP ;
- temps de cycle de 5 ns ;
- bus d'adresse de 32 bits ;
- format de données 32 bits.

**Mémoire/périphérique :**

- ✓ 64K octets L1P de mémoire cache de programme,
- ✓ 64K octets L1D de mémoire cache de données,
- ✓ 64 K octets L2 de mémoire cache RAM.
- 32 bits interface de mémoire externe (EMIF) ;
- contrôleur d'accès mémoire direct (EDMA) ;
- un port parallèle de 16 bits (HPI) ;
- 2 bus série (McBSP) ;
- 2 timers de 32 bits ;
- générateur d'horloge par PLL.

**4. Architecture interne du TMS320C6701**

Les unités fonctionnelles au niveau de la CPU figure (3.10), exécutent des opérations de logique, de décalage, de multiplication, etc. Les deux unités (.D1 et .D2) sont responsables de tous les transferts de données entre les registres dossiers et les mémoires. Les quatre unités fonctionnelles du même chemin de données ont un seul bus de données relié aux registres de l'autre côté de l'unité centrale de traitement, de sorte que ces unités puissent échanger les données avec le registre dossier du côté opposé .

**- Unité centrale de traitement (CPU)**

L'unité centrale de traitement contient :

- Unité de recherche de programmes ;
- Unité d'expédition d'instructions ;
- Unité de décodage d'instructions ;
- 32 registres de 32 bits ;
- Deux chemins de données, chacune avec quatre unités fonctionnelles ;
- Registres de contrôle ;
- Logique de contrôle ;

- Logique de test, émulation, et interruption.

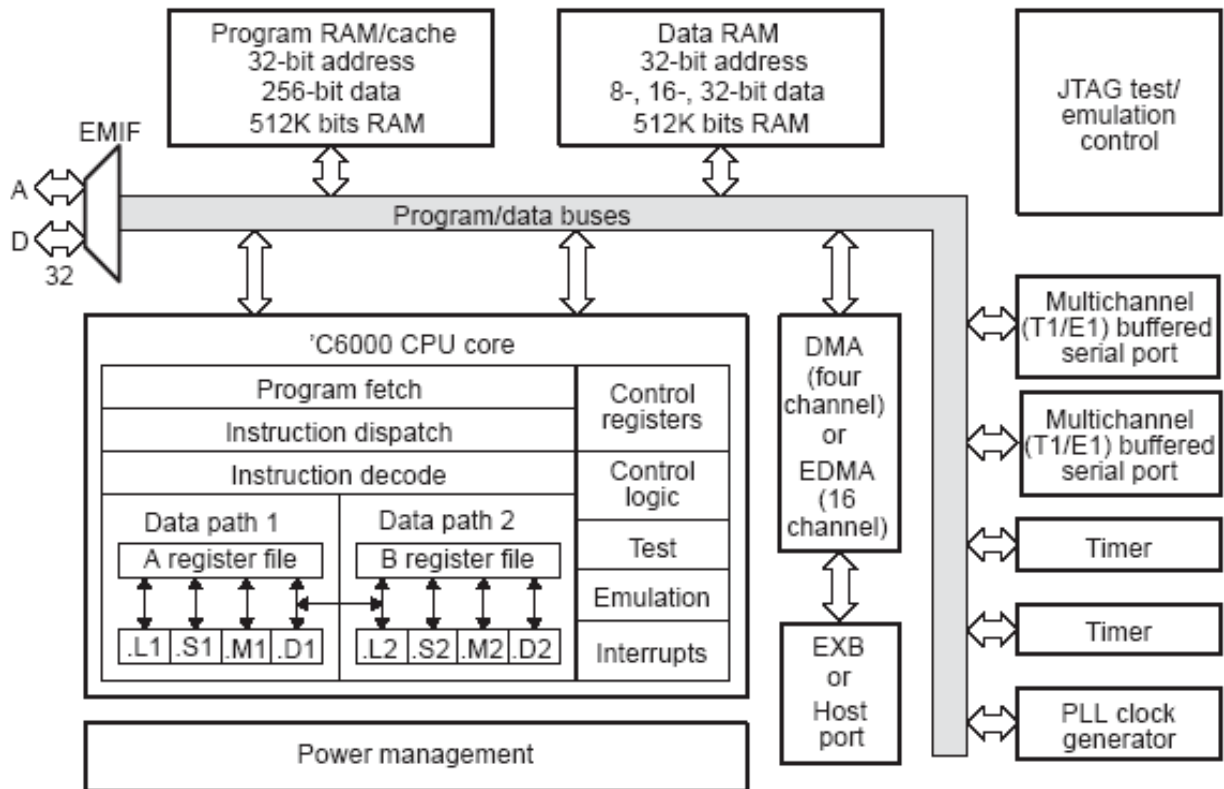


Figure 5.1 : Schéma fonctionnel du TMS320C67x

### - Les Mémoires internes

Le C6701 a une architecture basée en partie sur les mémoires caches. Des mémoires de niveau 1, séparées en mémoires programme et données (L1P et L1D). Ces espaces sont accessibles à tout moment par l'unité centrale de traitement. Le contrôleur de la mémoire cache de programme (L1P) interface l'unité centrale de traitement au L1P. Un chemin de 256-bits forme un jet (stream) continu de 8 instructions de 32 bits pour une exécution maximum. Le contrôleur de la mémoire cache de données (L1D) fournit l'interface entre l'unité centrale de traitement et L1D. Ce dernier permet l'accès simultané par les deux côtés de l'unité centrale de traitement. Quand un manque apparaît au niveau de L1D ou de L1P, la demande passe au contrôleur L2. Ce contrôleur facilite les fonctions suivantes :

- L'arbitrage nécessaire à l'accès mémoire interne entre CPU et contrôleur EDMA.
- L'accès des données d'unité centrale de traitement à l'EMIF.
- Les accès d'unité centrale de traitement aux périphériques sur chip.
- Envoi des demandes à EMIF pour des données de L2 manquantes.

## - Les registres

### \* Registres dossiers à usage général

Il y'a deux registres dossiers à usage général (A et B) dans les chemins de données du C6000. Pour les DSPs C62x/C67x, chacun de ces registres dossier contient 16 registres de 32 bits (A0–A15 pour le registre dossier A et B0–B15 registres dossier B) .

### \* Registres de contrôle

Les tableaux 3.2 a, b donnent une description des registres de contrôle de la famille C6000 et particulièrement celle de C67x.

Abréviation	Nom De Registre	Description
<b>AMR</b>	Addressing mode register	Indique si l'adressage utilisé est linéaire ou circulaire pour chacun des huit registres. Il contient également des tailles pour l'adressage circulaire.
<b>CSR</b>	Control status register	Contient le bit d'autorisation d'interruption global, bits de contrôle de la mémoire cache, et d'autres types de contrôle et bits d'état.
<b>PCE1</b>	Program counter, E1 phase	Contient l'adresse du paquet à chercher qui est dans l'étape de pipeline E1.
<b>FADCR</b>	Floating-point adder configuration registre	Spécifie le mode underflow, mode d'arrondissement, NaNs, et d'autres exceptions pour l'unité L.
<b>FAUCR</b>	Floating-point auxiliary configuration register	Spécifie le mode underflow, mode d'arrondissement, NaNs, et d'autres exceptions pour l'unité S.
<b>FMCR</b>	Floating-point multiplier configuration register	Spécifie le mode underflow, mode d'arrondissement, NaNs, et d'autres exceptions pour l'unité M.

Tableau.5.1 : Registres de contrôle de la famille C67x.

## 5. Organisation de la mémoire

Le tableau 5.2 donne la cartographie mémoire du TMS320C6701. Sur le kit d'évaluation, la zone GE0 est occupée par 16Mo de SRAM, la zone GE1 par 182Ko de ROM et de port d'entrée sortie.

Gamme d'Adresse	Taille (H)	Description du block mémoire
0000 0000 – 0000 FFFF	64K	RAM Interne (L2)
0001 0000 – 017F FFFF	24M–64K	Réservé
0180 0000 – 0183 FFFF	256K	Internal configuration bus EMIF registers
0184 0000 – 0187 FFFF	256K	Internal configuration bus L2 control registers
0188 0000 – 018B FFFF	256K	Internal configuration bus HPI register
018C 0000 – 018F FFFF	256K	Internal configuration bus McBSP 0 registers
0190 0000 – 0193 FFFF	256K	Internal configuration bus McBSP 1 registers
0194 0000 – 0197 FFFF	256K	Internal configuration bus timer 0 registers
0198 0000 – 019B FFFF	256K	Internal configuration bus timer 1 registers
019C 0000 – 019F FFFF	256K	Internal configuration bus interrupt selector registers
01A0 0000– 01A3 FFFF	256K	Internal configuration bus EDMA RAM and registers
01A4 0000– 01FF FFFF	6M–256K	Réserve
0200 0000 – 0200 0033	52	Registres de QDMA
0200 0034 – 2FFF FFFF	736M–52	Réserve
3000 0000 – 3FFF FFFF	256M	McBSP 0/1 data
4000 0000 – 7FFF FFFF	1G	Réserve
8000 0000 – 8FFF FFFF	256M	Interface de mémoire externe CE0
9000 0000 – 9FFF FFFF	256M	Interface de mémoire externe CE1
A000 0000–AFFF FFFF	256M	Interface de mémoire externe CE2
B000 0000– BFFF FFFF	256M	Interface de mémoire externe CE3
C000 0000– FFFF FFFF	1G	Réserver

Tableau.5.2 : Cartographie de la mémoire du C6701

## 6. Les périphériques

Les processeurs C6701 disposent de plusieurs périphériques intégrés auxquels sont associés des registres de contrôle mappés en mémoire. Les périphériques disponibles dépendent de la version du circuit et comprennent :

- Interface de mémoire externe (EMIF),
- un contrôleur d'accès mémoire direct (EDMA) à 17 canaux,
- 02 Timers de 32 bits,
- 02 Ports série (McBSP),

- un générateur d'horloge par PLL,

### - Le contrôleur EDMA (Enhanced Direct Memory Access)

Le TMS320C6701 élabore des transferts de données depuis un élément interne ou externe, en utilisant soit la CPU ou l'EDMA Figure 5.2. Typiquement les transferts depuis les périphériques internes au DSP sont faits par L'EDMA, libérant ainsi la CPU pour des tâches de calcul.

L'EDMA inclut plusieurs perfectionnements par rapport au DMA classique, vu le nombre canaux fournis avec priorités programmables , et la capacité de lier et enchaîner des transferts de données. L'EDMA permet le mouvement de données vers et depuis tous les espaces mémoires accessibles, y compris la mémoire interne (L2 SRAM), les périphériques, et la mémoire externe. Il prévoit deux types de transferts des données, à savoir, les transferts à une dimension (1D) et à deux dimensions (2D).

Les événements EDMA sont capturés dans le registre d'événements (Un événement est un signal de synchronisation qui déclenche un canal EDMA pour commencer un transfert). Si les événements se produisent simultanément, ils sont résolus par l'encodeur d'événement.

L'EDMA a la capacité d'exécuter vite les transferts effectifs en acceptant une demande de transfert DMA rapide (QDMA) de la CPU. Un transfert QDMA convient le mieux pour les applications qui exigent des transferts rapides de données tels que les demandes de données dans les algorithmes à boucles.

### - Les temporisateurs

Le TMS320C6701 a deux temporisateurs (timers) à usage général qui ont pour objectifs:

- 1) La mesure la durée d'un événement.
- 2) Le comptage des événements.
- 3) La génération des interruptions vers la CPU.
- 4) L'envoi des événements de synchronisation à l'EDMA.

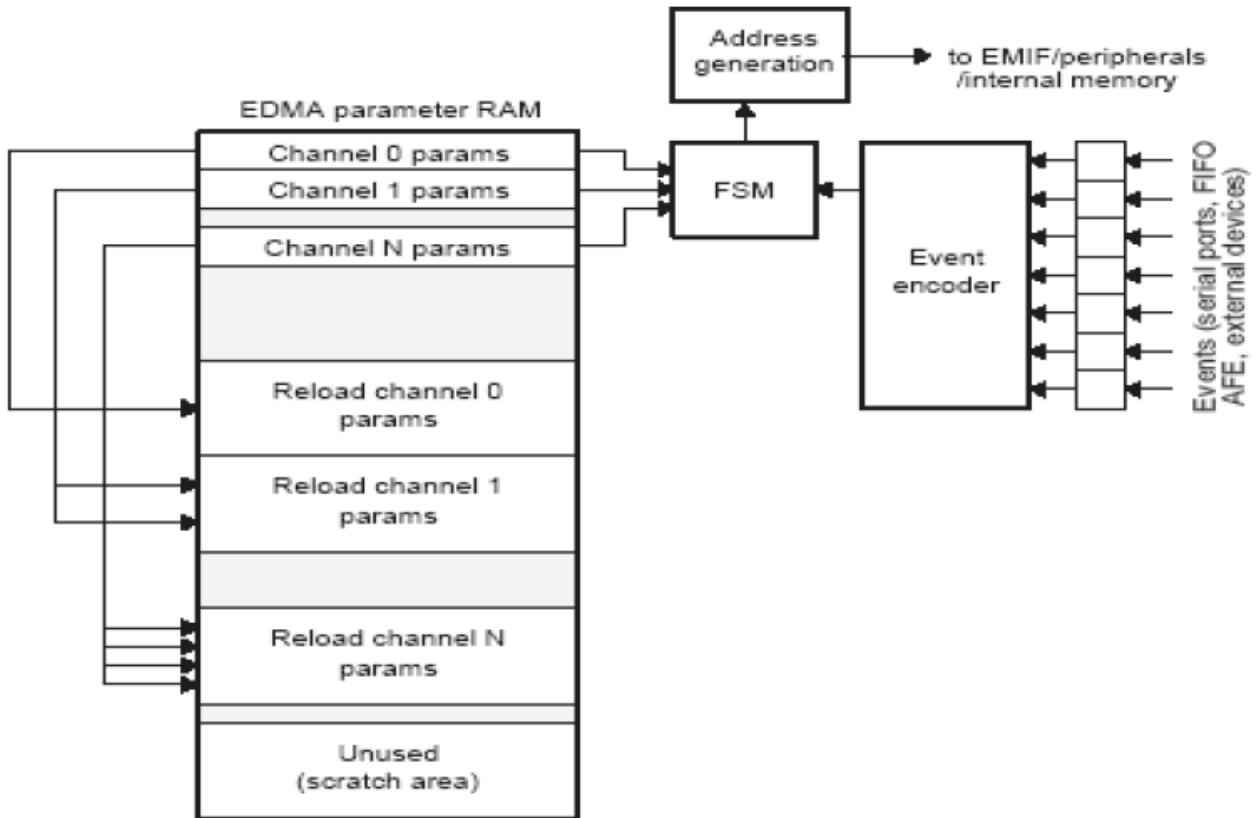


Figure.5.2 : Le contrôleur EDMA

Le contrôleur d'EDMA comporte :

- 1) Registres d'événements et de traitement d'interruptions
- 2) Encodeur d'événements
- 3) RAM paramètres
- 4) Hardware de génération d'adresses

#### - La liaison série (McBSP) (multichannel buffered serial port)

Le C6701 contient deux modules de liaison série (McBSP0 et McBSP1). Ces liaisons série permettent des communications en full-duplex, un flot continu des données, une indépendance dans le timing et le format des données à l'émission et la réception, une interface directe avec les CODEC, CNA et CAN, et une synchronisation par horloge interne ou externe *figure 5. 3*.

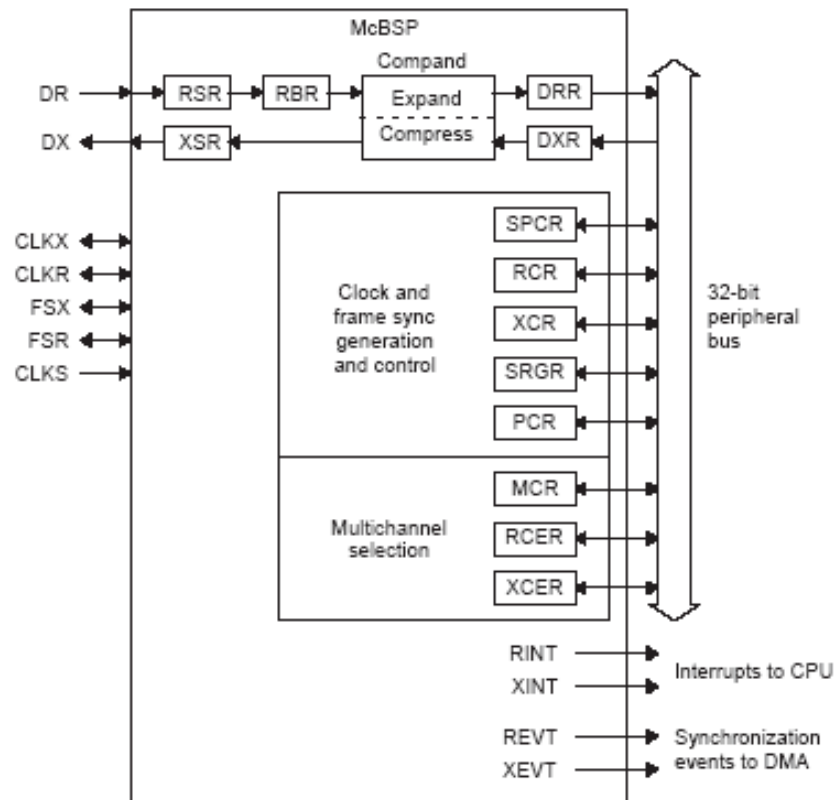


Figure.5.3 : Liaison série (McBSP)

### - Contrôleur PLL

Le contrôleur PLL est capable de générer différentes horloges pour différentes parties du noyau de DSP, l'interface externe de mémoire, McBSP, ériphériques, et d'autres modules à l'intérieur du DSP.

## 7. Jeu d'instructions

Le TMS320C6701 présente une liste très riche en instructions utilisées dans sa programmation en assembleur. On se limite de décrire quelques instructions.

### - Instructions de Addition

- Instructions pour les entiers

\* L'instruction **ADD**

**Syntaxe ADD** (.unit) src1, src2, dst

Ou **ADDU** (.L1 ou .L2) src1, src2, dst

Ou **ADD** (.D1 ou .D2) src2, src1, dst  
.unit = .L1, .L2, .S1, .S2

Exemple: ADD .L2X A1, B1, B2

\* L'instruction **ADDAB**

**Syntaxe ADDAB** (.unit) src2, src1, dst  
.unit = .D1 ou .D2

- Addition de réelle double précision

**Syntaxe ADDDP** (.unit) src1, src2, dst  
.unit = .L1 or .L2

Exemple: ADDDP .L1X B1:B0, A3: A2, A5: A4

## - Instructions de multiplication

- Instructions pour les entiers

**Syntaxe MPY** (.unit) src1, src2, dst

Ou **MPYU** (.unit) src1, src2, dst  
.unit = .M1 ou .M2

Exemple : MPYU .M1 A1, A2, A3

- Addition de réelle double précision

**Syntaxe MPYDP** (.unit) src1, src2, dst  
.unit = .M1 ou .M2

Exemple : MPYDP .M1 A1:A0, A3:A2, A5:A4

## 8. Modes d'adressages

Les modes d'adressage sur le C671 sont : linéaire et circulaire en utilisant BK0 ou BK1. Le mode est indiqué par le registre de mode d'adressage, ou (AMR). Tous les registres peuvent exécuter l'adressage linéaire. Seulement huit registres peuvent exécuter l'adressage circulaire (A4–A7 sont employés par l'unité .D1, et B4–B7 employés par l'unité .D2). Aucune autre unité ne peut exécuter l'adressage circulaire.