CHAPITRE 2 : MICROCONTROLEURS ET MICROPROCESSEURS EMBARQUES

2.1 Introduction

Un processeur embarqué est un microprocesseur utilisé dans un système embarqué. Ces processeurs sont généralement plus petits, utilisent un facteur de forme de montage en surface et consomment moins de puissance. Les processeurs intégrés peuvent être divisés en deux catégories : microprocesseurs ordinaires et microcontrôleurs. Les microcontrôleurs ont plus de périphériques sur la puce. Essentiellement, un processeur embarqué est une puce CPU utilisée dans un système qui n'est pas une station de travail, un ordinateur portable ou un ordinateur de bureau à usage général.

2.2 Les processeurs ARM

La société ARM s'appelait à l'origine « Acorn Computer ». Cette société anglaise était spécialisée au début des années 1980 dans le développement de micro-ordinateurs. Son succès est en grande partie dû à la commercialisation de l'ordinateur familial BBC.

Les processeurs de la famille ARM sont tous conçus autour d'un cœur de processeur standard de type RISC. Ce cœur de processeur a été conçu de manière à obtenir des implémentations matérielles simples et efficaces, autorisant ainsi des performances importantes avec de faibles consommations électriques.

Ils sont donc caractérisés par :

1. une taille fixe des instructions (32 bits),
2. une banque de registres (32 bits) à usages généraux,
3. un décodage des instructions câblé (pas de décodage par microcode ou microprogramme),
4. une exécution pipelinée (trois étages pour les ARM7, cinq pour les ARM9, six pour les ARM10 et huit pour les ARM11),
5. un objectif d'exécution d'un cycle par instruction (CPI = 1).

Le standard ne définissant que la structure et le jeu d'instructions du cœur de processeur, chaque industriel est susceptible de compléter son processeur avec des modules matériels spécifiques selon la destination finale du processeur.

Il existe donc un certain nombre d'extensions pouvant optionnellement être adjointes au coeur. On trouve par exemple:

1. *Le jeu d'instructions Thumb* : il s'agit d'un sous-ensemble du jeu d'instructions standard, codé sur un format réduit.
2. Des instructions de multiplication longues, susceptibles de produire des résultats sur 64 bits.
3. Des instructions spécifiques au traitement du signal.
4. Des modules de débogage avancés pour tracer pas à pas l'exécution des applications.

2.3 Architecture interne et jeu d'instructions

Les cœurs de processeurs ARM partagent une architecture commune qui peut être résumée comme suit :

- Processeurs RISC (Reduced Instruction Set Computer) :
  - Banc de registres uniforme indifférenciés.
  - Architecture « load/store » : seules les instructions de chargement et rangement accèdent à la mémoire ; toutes les autres instructions (traitements) travaillent sur les registres.
  - Modes d'adressages réduits, opérant uniquement à partir des valeurs des registres et de décalages.
  - Instructions de longueur fixe et de structure uniforme afin d'accélérer le décodage.
- Possibilité d'utiliser l'unité arithmétique et logique pour chaque instruction de traitement.
- Gestion des modes d'adressage auto-incrémentés et auto-décrémentés.
- Gestion des instructions de chargement et rangement par blocs.
- Exécution conditionnelle de toutes les instructions.

Ces caractéristiques garantissent des performances maximales de consommation et de complexité du processeur.

2.3.1 Architecture interne

Le détail de la structure interne du cœur de processeur ARM7 est représenté sur la figure 2.1. On y trouve le pipeline et ses trois étages, la banque de registres, l'unité arithmétique et logique et les divers registres intermédiaires.

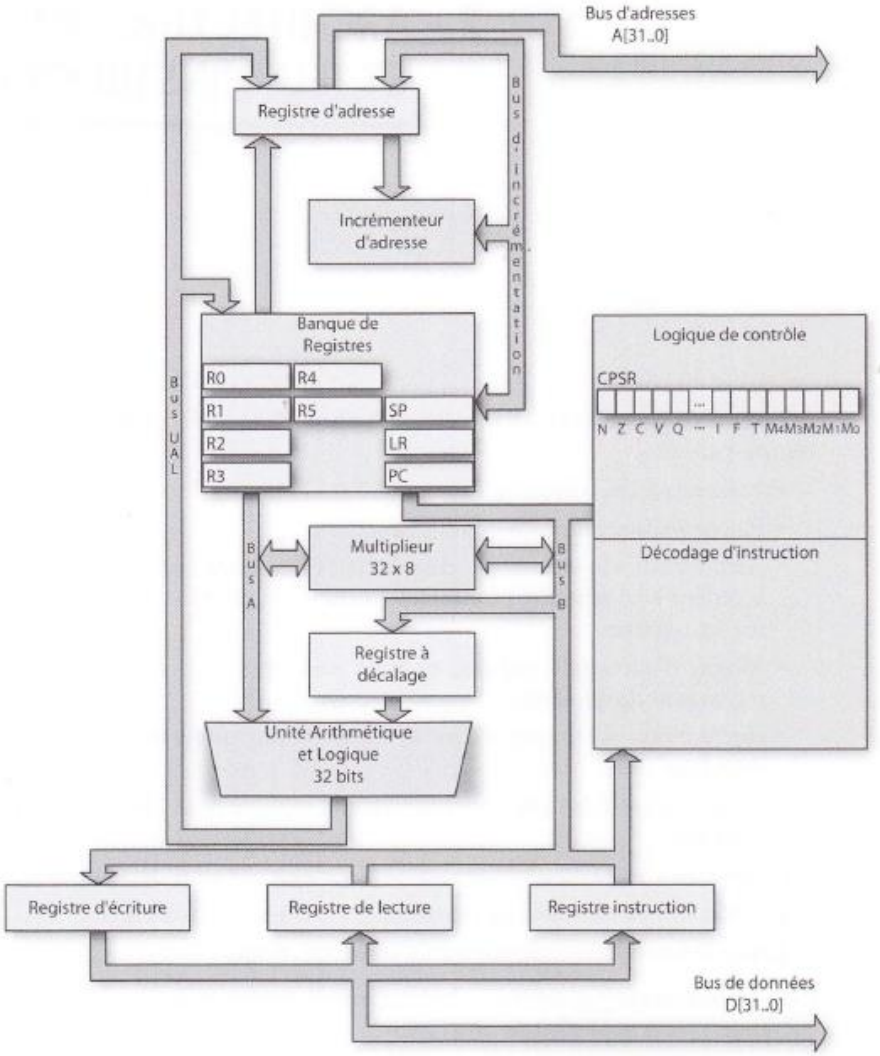

Figure 2.1 Structure Interne d'un cœur de processeur ARM7.

2.3.2 Organisation de la mémoire et des entrées/sorties

L'espace mémoire d'un processeur ARM est constitué de 2^{32} octets. Les adresses s'étendent donc de 0 à 2^{32}-1. Notons que le processeur accède à la mémoire par le biais de deux bus: le bus d'adresse (en haut sur le schéma) et le bus de données (en bas).

Le bus d'adresse est utilisé par le processeur pour dialoguer avec la mémoire. Pour tout transfert, le processeur déposera dans le registre d'adresse l'adresse relative à ce transfert. Cela concerne donc aussi bien l'accès aux instructions (l'étage de Lecture instruction du pipeline) que l'accès aux données en lecture ou en écriture (les chargements et les rangements mémoire).

Le bus de données donne l'accès à trois registres: deux pour l'échange des données entre le processeur et la mémoire, (écriture et lecture) permettent de mettre en œuvre les transferts entre la banque de registres et la mémoire. Le registre d'instructions constitue le premier étage du pipeline d'exécution de l'ARM.

2.3.3 Types de données

Les types de données manipulables par les processeurs ARM sont les suivants :

Word: mot. Un mot est une donnée codée sur 32 bits, soit 4 octets. Il doit impérativement être aligné sur une adresse divisible par 4. La mémoire étant adressable à l'octet, un mot est donc rangé dans les octets d'adresse A, A+1, A+2 et A+3 (A étant donc divisible par 4).

Halfword : demi-mot. Un demi-mot est une donnée codée sur 16 bits, soit 2 octets. Il doit impérativement être aligné sur une adresse paire. Un demi-mot sera donc rangé dans les octets d'adresse A et A+l, A étant pair.

Byte: octet. Un octet est une donnée codée sur 8 bits. Il peut être rangé n'importe où en mémoire.

Little-endian ou big-endian

La spécification des processeurs ARM n'impose pas l'utilisation d'une organisation big-endian ou litcle-endian. par le biais d'un signal extérieur que la sélection se fait (signal BIGEND : à l, indique que le système mémoire est big-endian, à 0 indique que le système mémoire est little-endian).

A. **Big-endian :** les bits de plus fort poids du mot sont stockés à l'adresse la plus basse. Le mot est donc accédé par l'adresse de l'octet qui contient les bits les plus significatifs du mot.

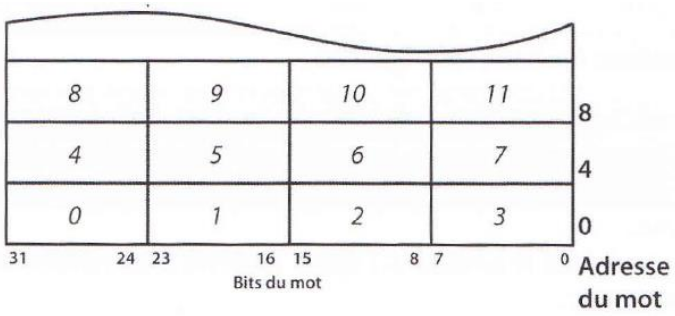

Figure 2.2 Organistion Big endian.

B. **Little-endian :** les bits de plus faible poids du mot sont stockés à l'adresse la plus basse. Le mot est donc accédé par l'adresse de l'octet qui contient les bits les moins significatifs du mot.

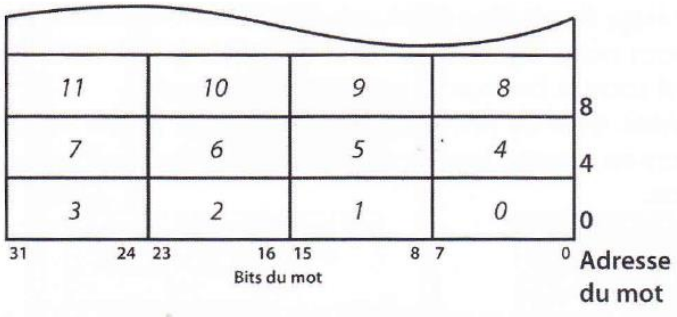

Figure 2.3 Organistion little endian.

2.3.4 Organisation du banc de registres

Le banc de registre est constitué de 31 registres généraux 32 bits et six registres d'état. Cependant, à chaque instant, seuls 16 registres généraux et deux registres d'état sont visibles; les autres sont masqués selon le mode de fonctionnement du processeur.

On a résumé dans le tableau 2.1 quels registres sont visibles suivant quel mode de fonctionnement.

Registres généraux

Les 16 registres généraux visibles à chaque instant sont utilisables dans toute instruction de l'ARM. Ils sont presque tous identiques ; deux seulement ont des rôles particuliers :

R14 : Link Register - Ce registre contient l'adresse de l'instruction qui suit l'exécution d'une instruction branch and link (BL) (interruption). Cette dernière est utilisée pour appeler un sous-programme. Il peut être utilisé comme n'importe quel autre registre.

RI5: Program Counter - Il s'agit du compteur ordinal, pointant à chaque instant, deux instructions après l'instruction en cours d'exécution, puisque la profondeur du pipeline d'exécution est (3).

Registres d'état

Il existe deux types de registres d'état : le CPSR, accessible dans tous les modes, et le SPSR accessible uniquement dans les modes d'exception. Ce dernier n'est qu'une copie du CPSR lorsqu'une exception survient.

Le CPSR stocke deux types d'informations: les bits de code condition et les bits de contrôle.

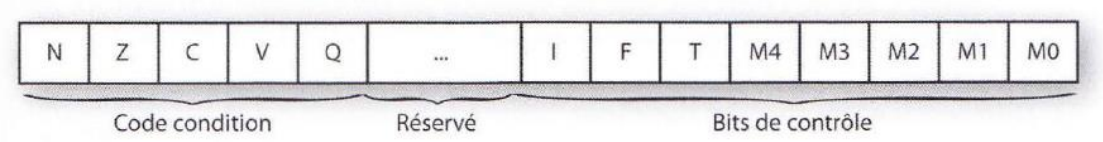

Figure 2.4 - Le registre CPSR.

Bits de code condition

Les cinq bits N, Z, C, V et Q décrivent le résultat de l'exécution d'opérations de comparaisons ou de manipulation. La modification du code condition a pour objectif de savoir si le résultat de l'exécution de l'instruction a produit un nombre négatif, nul, un dépassement, etc.

Les différents bits

La signification des différents bits de code condition est la suivante:

- **N (Negative)** : reflète l'état du bit 31 du résultat de l'instruction. Si ce résultat doit être interprété comme un complément à deux, ce bit rend compte du signe du résultat (1 : négatif, 0 : positif).
- **Z (Zero)** : bit à 1 lorsque le résultat de l'instruction est 0, à 0 sinon.

- **C (Carry) :** représente la retenue à propager.
- **V (oVerflow)** : dépassement de capacité. V est positionné à 1 si un dépassement se produit à la suite d'une opération d'addition ou de soustraction,
- **Q** : dépassement de capacité ou saturation dans les opérations de traitement du signal.

Bits de contrôle

Il s'agit des huit bits de faible poids du CPSR. Ces bits peuvent être modifiés de deux manières :

- lorsqu'une exception se produit ;
- par une modification explicite par programme. Ceci n'est possible que lorsque le processeur est en mode privilégié.

La signification de ces différents bits est la suivante:

- **I et F** : ces deux bits permettent de désactiver les interruptions IRQ et FIQ respectivement. Il faut, pour cela, mettre le bit correspondant à 1.
- **T** : indique le jeu d'instructions utilisé par le processeur. Lorsqu'il est égal à 0, le jeu d'instructions utilisé est le jeu standard de l'ARM. Lorsqu'il est à l, le jeu d'instructions utilisé est le jeu *Thumb*, ce qui suppose que le processeur possède cette option et qu'une instruction permettant de changer de mode a été appelée. Lorsque l'option *Thumb* n'est pas disponible sur le processeur, ce bit doit toujours être à 0.
- **M (Mode)** : M[4 .. 0] permettent de déterminer le mode de fonctionnement du processeur. La signification de ces bits est donnée ci-dessous.

Tableau 2.2 - Valeurs du champ Mode du registre CPSR.

M[4..0]	Mode
0b10000	User
0b10001	Fast Interrupt (FIQ)
0b10010	Interrupt (IRQ)
0b10011	Supervisor
0b10111	Abort
0b11011	Undefined
0b11111	System

Remarque : Seules ces combinaisons sont des combinaisons valides. Si une autre combinaison est écrite dans le registre, le résultat est non prévisible.

2.3.5 Le pipeline d'exécution et les unités fonctionnelles

Il s'agit de la partie opérative du processeur.

Nous allons présenter son organisation générale, et préciser les différents modes de fonctionnement du processeur.

Organisation interne

Le pipeline d'exécution, situé à droite sur la figure 2.1, est représenté par:

- Le registre instruction, qui représente l'étage de Lecture instruction. (*Fetch*)
- La logique « décodage d'instruction », qui représente l'étage de décodage. (*decode*)
- La logique de contrôle qui représente l'étage d'exécution. (*Execute*)

1- l'adresse présente dans le registre d'adresse est déposée sur le bus d'adresse pour obtenir de la mémoire l'instruction suivant celle qui est en cours de décodage.
2- L'instruction obtenue de la mémoire est alors écrite dans le registre instruction ; elle sera transférée dans la logique de décodage lors du cycle suivant.
3- La logique de contrôle est en charge de l'exécution de l'instruction. Elle sélectionne l'opération à effectuer et les registres opérandes source, ainsi que le registre opérande destination.

2.3.6 Modes de fonctionnement du processeur et exceptions

User (usr) : mode de fonctionnement normal.

Fast Interrupt (fiq) : mode de traitement des interruptions avec support de transfert de données amélioré.

Interrupt (irq) : traitement standard, général, des interruptions.

Supervisor (svc) : mode protégé, dédié à l'exécution du système d'exploitation.

Abort (abt) : mode de gestion des défauts mémoire,

Undefined (und) : mode utilisé pour l'émulation de coprocesseur.

System (sys) : mode utilisé par le système d'exploitation pour effectuer des tâches nécessitant des privilèges étendus.