

Jacques Jorda

# **P**ROCESSEURS ARM

**Architecture  
et langage d'assemblage**

DUNOD

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1<sup>er</sup> juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, Paris, 2010

ISBN 978-2-10-056073-8

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

# SOMMAIRE

---

Avant-propos

VII

## A

---

### Cœurs de processeurs ARM : architecture et programmation

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>1 • Les versions de cœurs de processeurs</b>           | <b>3</b>  |
| 1.1 Historique                                            | 3         |
| 1.2 Caractéristiques fondamentales                        | 4         |
| 1.3 Versions et implémentations                           | 5         |
| <b>2 • Architecture interne et jeu d'instructions</b>     | <b>11</b> |
| 2.1 Architecture interne                                  | 11        |
| 2.2 Présentation du jeu d'instructions standard de l'ARM7 | 21        |
| 2.3 Codage des instructions et exécution                  | 29        |
| <b>3 • Techniques de programmation assembleur</b>         | <b>43</b> |
| 3.1 Structures algorithmiques classiques                  | 43        |
| 3.2 La mémoire et les tableaux                            | 46        |
| 3.3 Utilisation des sous-programmes                       | 62        |
| 3.4 Gestion des exceptions                                | 69        |

## B

---

### Les entrées/sorties

|                                                            |           |
|------------------------------------------------------------|-----------|
| <b>4 • Les entrées/sorties programmées par test d'état</b> | <b>87</b> |
| 4.1 Concepts généraux                                      | 87        |

|                                                                |            |
|----------------------------------------------------------------|------------|
| 4.2 Implémentation des entrées/sorties : la famille Atmel AT91 | 95         |
| 4.3 Le PIO ( <i>Parallel Input Output Port</i> )               | 99         |
| 4.4 Les timers                                                 | 110        |
| 4.5 Les liaisons séries                                        | 126        |
| <b>5 • Les entrées/sorties par interruptions</b>               | <b>139</b> |
| 5.1 Entrées/sorties programmées par interruptions              | 139        |
| 5.2 L'AIC ( <i>Advanced Interrupt Controller</i> )             | 142        |
| 5.3 Les interruptions du timer                                 | 160        |

## C

### Annexes

|                                                         |            |
|---------------------------------------------------------|------------|
| <b>A • Jeu d'instructions de l'ARM</b>                  | <b>171</b> |
| A.1 Exécution conditionnelle                            | 171        |
| A.2 Instructions de branchement                         | 173        |
| A.3 Instructions de traitement des données              | 175        |
| A.4 Instructions de chargement/rangement                | 194        |
| A.5 Instructions diverses                               | 205        |
| <b>B • Résumé du jeu d'instructions de l'ARM 7 TDMI</b> | <b>217</b> |
| <b>C • Exercices</b>                                    | <b>221</b> |
| C.1 Climatisation                                       | 221        |
| C.2 Système d'arrosage programmable                     | 228        |
| C.3 Jeu de cache-cache                                  | 237        |
| C.4 Correcteur automatique de QCM                       | 246        |
| C.5 Radar de recul                                      | 251        |
| <b>Index alphabétique</b>                               | <b>259</b> |

## ■ Pourquoi les processeurs ARM ?

Avec déjà 5 milliards d'unités vendues, les processeurs basés sur les cœurs ARM ont envahi notre quotidien. On les retrouve dans des appareils aussi variés que des téléphones mobiles, des imprimantes, des équipements réseau, des appareils multimédias et des consoles de jeu, des appareils photos, des ordinateurs de voiture, etc. Ceci s'explique par un rapport performance/consommation imbattable qui les rend particulièrement adaptés aux environnements embarqués.

## ■ Plan du livre

Nous débuterons par une étude générale des cœurs de processeurs ARM qui nous permettra de détailler la structure interne d'un cœur de processeur et de préciser le rôle de chaque unité fonctionnelle. Nous introduirons ensuite le jeu d'instructions (une liste exhaustive est fournie en annexe A) et nous terminerons par l'analyse des techniques de programmation assembleur classiques appliquées à l'ARM : structures algorithmiques classiques, modèles de gestion de la mémoire et de la pile, gestion des exceptions.

Dans un second temps, nous étudierons les mécanismes d'entrées/sorties, en distinguant les entrées/sorties par test d'état et les entrées/sorties par interruptions. Dans chaque cas, nous commencerons par introduire les mécanismes généraux, puis nous nous appuierons sur une famille de circuits spécifique commercialisée par Atmel (la famille AT91SAM7S) pour montrer les implémentations effectives. Bien que l'étude des différentes unités fonctionnelles s'applique aux circuits de la famille AT91SAM7S, ce contenu est facilement transposable à d'autres familles de microcontrôleurs basées sur des cœurs ARM7 (on pense notamment aux Philips LPC2000). En effet, les connaissances acquises ici permettront d'appréhender facilement les documents techniques de n'importe quelle autre plate-forme équivalente.

## ■ À qui s'adresse cet ouvrage ?

En présentant à la fois les aspects matériels et logiciels des processeurs ARM, cet ouvrage permettra aux professionnels qui utilisent ces processeurs de comprendre les enjeux et les contraintes de ces architectures.

Il est aussi parfaitement indiqué pour les étudiants de licence qui y trouveront à la fois l'aspect théorique des principes fondamentaux et de nombreux exemples de mise en œuvre.

Enfin, l'amateur souhaitant, dans le cadre de projets domotiques, robotiques ou autres, s'initier à l'utilisation d'une famille de processeurs garantissant faible coût et performances élevées, trouvera ici le substrat nécessaire à de nombreuses réalisations.

## ■ Remerciements

Cet ouvrage n'aurait pas vu le jour sans la patience, les encouragements et le soutien de ma famille, ni sans le dévouement d'Abdelaziz M'zoughi qui a relu et corrigé le manuscrit.

Merci infiniment.

# A

---

## Cœurs de processeurs ARM : architecture et programmation



# 1 • LES VERSIONS DE CŒURS DE PROCESSEURS

---

Le processeur ARM 7 TDMI appartient à une famille très complète et très répandue de processeurs. Nous allons présenter l'historique de cette famille et les caractéristiques fondamentales des modèles qui la composent. Nous passerons ensuite en revue les différentes versions du cœur de processeur et les implémentations correspondantes.

## 1.1 Historique

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. Cet ordinateur, commandé à Acorn Computer par la BBC (*British Broadcasting Corporation*), fut commercialisé dès 1982. Il était équipé d'un 6502 (processeur 8 bits de Rockwell qui équipait déjà l'Apple II) et proposait des graphiques en couleur, 32 Ko de mémoire vive et un système de sauvegarde sur cassettes (en option, disquette et disque dur).

En 1983, Steve Furber, Roger Wilson et Robert Heaton commencèrent à travailler sur la première version de l'ARM : l'ARM1. Ils avaient comme objectif de créer un processeur 32 bits de type RISC (*Reduced Instruction Set Computer*) dans la lignée du 6502. La principale décision fut donc de s'appuyer sur des instructions de longueur fixe et un modèle load/store (seules les opérations de lecture et d'écriture accèdent à la mémoire, les opérations de traitement ne travaillant que sur les registres). L'ARM1 avait un jeu d'instructions en BASIC. Son architecture matérielle était elle aussi écrite en BASIC (l'implémentation physique VLSI fut réalisée par un outil spécifique), de même qu'un simulateur de ce processeur (ce dernier, appelé ASIM, fut par la suite réécrit en Modula-2 puis en C). La première implémentation physique a vu le jour en avril 1985 dans les laboratoires de VLSI Technology ; elle comportait moins de 25 000 transistors gravés à 3  $\mu\text{m}$ . Il s'agissait d'un des premiers processeurs RISC, et il ne fut produit qu'à quelques centaines d'exemplaires.

L'ARM1 fut amélioré pour accroître les performances des systèmes cibles. Il lui fut donc ajouté des instructions de multiplication et de multiplication-addition, prépondérantes en traitement du signal (donc pour la production de son, caractéristique

importante pour les ordinateurs familiaux). Il lui fut aussi adjoint une interface coprocesseur (notamment pour le traitement des calculs flottants). Le processeur correspondant, ARM2, parvint à maintenir les objectifs de faible surface et faible nombre de transistors.

La crise financière de 1985 frappa Acorn Computer de plein fouet, et la société tomba dans le giron d'Olivetti. La commercialisation du premier ordinateur personnel équipé du processeur ARM2 prit du retard et le premier modèle n'apparut sur le marché qu'en 1987. L'Archimède était équipé d'un processeur ARM2 cadencé à 8 MHz. Il possédait son propre système d'exploitation, mais très peu de programmes. Ce dernier point handicapa lourdement son succès commercial face à l'IBM PC.

Les équipes de recherche et développement continuèrent leur travail en ajoutant un cache instruction et données de 4 Ko et en augmentant la fréquence du processeur pour atteindre 25 MHz. Les ordinateurs Acorn équipés de cette version du processeur furent commercialisés en 1990. Dans le même temps, la société produisit une version à consommation réduite du processeur ARM2. Ce processeur avait vocation à équiper les appareils électroniques portables à bas coûts.

La société ARM fut créée en novembre 1990, en groupant les énergies d'Apple, de VLSI Technologies et d'Acorn Computer. Les deux premières apportèrent des capitaux tandis qu'Acorn fournit son expertise dans les processeurs RISC. L'objectif de la société ARM Ltd était de définir un nouveau standard de processeurs qui devaient consommer peu, pouvoir être produits à bas coûts et utiliser le modèle RISC. Le premier développement de la société ARM fut l'ARM6, correspondant à la version de cœur de processeur 3 (!)<sup>1</sup> ; il utilisait un adressage 32 bits complet, et était proposé avec un coprocesseur flottant et un contrôleur vidéo.

Un tournant fut pris en janvier 1992. ARM signa ses premiers accords de licence avec un fondeur britannique (GEC Plessey Semiconductors). D'autres suivirent, et ARM Ltd est maintenant garante du maintien de la cohérence de familles entières de processeurs compatibles produits par un nombre impressionnant de partenaires industriels. La société emploie aujourd'hui plus de 700 personnes dans des bureaux répartis tout autour du globe.

## 1.2 Caractéristiques fondamentales

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. L'utilisation des processeurs de cette famille dans les périphériques mobiles s'est donc rapidement imposée : PDA, consoles de jeux portables, etc.

Les processeurs ARM étant des processeurs RISC, ils sont donc caractérisés par :

- une taille fixe des instructions (32 bits),
- une banque de registres (32 bits) à usages généraux,

---

1. Les numéros de versions de cœur de processeur diffèrent des appellations commerciales.

- un décodage des instructions câblé (pas de décodage par microcode ou microprogramme),
- une exécution pipelinée (trois étages pour les ARM7, cinq pour les ARM9, six pour les ARM10 et huit pour les ARM11),
- un objectif d'exécution d'un cycle par instruction (CPI = 1).

Cette simplicité des processeurs RISC, comparée à celle des processeurs CISC (*Complex Instruction Set Computer*) permet d'atteindre des performances accrues pour des surfaces de silicium plus réduites.

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, fonctions de la destination finale du processeur. Deux processeurs ARM de même version ne seront donc pas similaires, bien que les cœurs soient identiques et donc compatibles. Il existe donc un certain nombre d'extensions pouvant optionnellement être adjointes au cœur. On trouve par exemple :

- Le jeu d'instructions Thumb : il s'agit d'un sous-ensemble du jeu d'instructions standard, codé sur un format réduit. Les programmes utilisant ce jeu d'instructions sont donc plus compacts. Cependant, le jeu d'instructions Thumb étant plus élémentaire, il faut généralement plus d'instructions qu'avec le jeu complet pour effectuer la même tâche ; il en résulte que le jeu standard est utilisé lorsque des performances élevées doivent être obtenues.
- Des instructions de multiplication longues, susceptibles de produire des résultats sur 64 bits.
- Des instructions spécifiques au traitement du signal. Ces instructions comprennent entre autres des opérations sur 16 bits ainsi que des opérations en arithmétique saturée.
- Des modules de débogage avancés pour tracer pas à pas l'exécution des applications.

## 1.3 Versions et implémentations

À chaque version du cœur de processeur correspond une ou plusieurs implémentations physiques. On dénombre aujourd'hui six versions différentes du cœur.

### 1.3.1 Version 1

Cette version du cœur correspond au processeur ARM1. Seuls quelques centaines d'exemplaires ont été réalisés, uniquement à des fins de test et d'évaluation. Le processeur ARM1 sera rapidement remplacé par l'ARM2, qui ne présente que peu de différences avec son aîné.

### 1.3.2 Version 2

La version 2 du cœur a été implémentée par le processeur ARM2. Comparé au processeur ARM1, l'ARM2 possède 27 registres (dont 16 accessibles simultanément), et quatre modes de fonctionnement différents :

- **USR** (USeR mode) : mode utilisateur. Il s'agit du mode de fonctionnement des programmes utilisateurs. Certaines ressources sont inaccessibles (il s'agit du seul mode de fonctionnement non privilégié), et il est impossible de changer de mode autrement qu'en causant une interruption logicielle.
- **IRQ** (*Interrupt ReQuest mode*) : mode interruption (les registres R13 et R14 sont remplacés par des copies privées). Ce mode est utilisé pour gérer les interruptions matérielles externes : il devient actif lorsqu'un signal actif est reçu sur l'entrée IRQ du processeur.
- **FIQ** (*Fast Interrupt reQuest mode*) : mode interruption rapide (les registres R8 à R14 sont remplacés par des copies privées<sup>1</sup>). Ce mode est actif lors de la réception d'un signal sur l'entrée FIQ du processeur. Ce mode se distingue du précédent par des capacités supérieures en terme de transfert de données, et un nombre de registres privés supérieur. La priorité du mode FIQ est supérieure à celle du mode IRQ.
- **SVC** (*Supervisor mode*) : mode Superviseur (les registres R13 et R14 sont remplacés par des copies privées). C'est un mode protégé, utilisé pour l'exécution du système d'exploitation.

Dans cette version, les 6 bits de fort poids du compteur de programme sont utilisés pour stocker l'état du processeur ; en conséquence, l'adressage est limité à 26 bits sur ces versions (en réalité, 8 bits sont utilisés comme registre d'état, les deux bits supplémentaires viennent du fait que le compteur de programme manipule des adresses octets, les instructions étant codées sur 4 octets ; par conséquent, les 2 bits de plus faible poids sont toujours à 0).

Le pipeline d'exécution est constitué de trois étages : Lecture instruction, Décodage instruction et Exécution instruction. Les instructions peuvent être exécutées de façon conditionnelle, et les données peuvent être traitées préalablement par des opérations de rotation, de décalage logique ou de décalage arithmétique.

Les différentes classes d'instructions exécutables sont :

- **Accès mémoire** : chargement et rangement d'un ou plusieurs registres, sur des opérandes de taille mot (32 bits), demi-mot ou octet (load/store).
- **Transfert/affectation** : transfert d'un registre ou de son complément dans un autre registre, affectation d'une constante ou de son complément à un registre (Move/Move NOT).
- **Arithmétiques** : additions avec ou sans retenues, soustraction avec ou sans retenues, soustraction inverse avec ou sans retenues.
- **Comparaison arithmétique et test logique** : Compare/Compare Negative.
- **Logique booléenne** : tests, test d'équivalence, ET, OU, OU Exclusif, Bit Clear.
- **Contrôle de flux de programme** : Branchements conditionnels et inconditionnels, Branchement avec lien.
- **Interruption logicielle.**
- Pour la version 2 seulement :
  - plus de spécification de registres dans les load/store,
  - multiplication et multiplication – addition,

1. Dans la version 1, seuls les registres R10 à R14 sont remplacés par des copies privées.

- instructions de contrôle d'un coprocesseur : opérations de données, échange entre le coprocesseur et les registres du processeur.

L'ARM2 produit industriellement fonctionnait à une fréquence de 8 MHz, pour une performance de 4 à 5 MIPS.

### 1.3.3 Version 2a5

Version 2 améliorée par ajout d'une instruction de swap (SWP), assignation au CPU du numéro de coprocesseur 0, et cache.

L'instruction de swap SWP permet l'échange de données monolithique et atomique entre un registre et la mémoire, et est utile dans l'écriture de programmes en environnement multiprocesseur.

Le cache est un cache unifié pour les instructions et les données de 4 Ko inclus dans la puce. Il s'agit d'un cache associatif par ensemble (64 lignes par ensemble, quatre mots par ligne) avec mise à jour immédiate et politique de remplacement aléatoire<sup>1</sup>.

Cette version de cœur a été mise en œuvre dans les processeurs ARM3 (26 MHz à 33 MHz) et ARM250 (12 MHz avec contrôleur vidéo, mémoire et E/S intégrés mais sans cache).

### 1.3.4 Version 3

Dans cette version, la limitation de l'espace d'adressage à 64 Mo est supprimée avec un véritable adressage 32 bits et un nouvel ensemble de registres est ajouté pour maintenir les informations concernant l'état du processeur :

- CPSR : *Current Processor Status Register*.
- SPSR : *Stacked Processor Status Register*.

Par voie de conséquence, quatre catégories d'instructions de transfert sont ajoutées :

- CPSR → Registre.
- SPSR → Registre.
- Registre → CPSR.
- Registre → SPSR.

Par ailleurs, deux nouveaux modes de fonctionnement sont ajoutés :

- **Abort** : signalé par le gestionnaire de la mémoire, utilisé pour implémenter des mécanismes de mémoire virtuelle et/ou de protection mémoire.
- **Undefined** : utilisé pour l'émulation logicielle de coprocesseur.

Cette version propose aussi le choix de gestion de la mémoire : little-endian et big-endian. À noter que par soucis de compatibilité, l'émulation du mode d'adressage 26 bits est proposée.

Cette version du cœur a été mise en œuvre dans les processeurs ARM6, ARM7 et Amulet1. L'ARM6 possède un certain nombre de dérivés incluant ou non des coprocesseurs directement dans la puce (Gestionnaire mémoire, cache modifié

1. Aucune précision sur la mise en œuvre exacte de cette politique dans les implémentations.