

ARM9やARM11のアーキテクチャを理解するには、ARMアーキテクチャの基本であるARM7から理解する必要があります。ここではARMアーキテクチャの基本から、各種アーキテクチャのバージョン、そしてARM9やARM11などの各種ARMファミリについて解説する。(編集部)

1 ARMプロセッサの概要

1.1 基本仕様

ARMプロセッサの基本的な仕様について具体的に説明します。ここで説明する仕様は、現時点(2010年3月)でのARMプロセッサすべてに共通です。

● 32ビットRISCプロセッサ

現在、ARMプロセッサはすべてのアーキテクチャが32ビット・プロセッサです。レジスタ長は32ビットで、ALUの演算長は32ビット、アドレス空間は32ビット・アドレスの4Gバイトです。

● 命令セット

命令には32ビットのARM命令セットと16ビットのThumb(サム、親指という意味)命令セットがあります。Thumb命令セットは32ビットのARM命令を16ビットに圧縮したサブセットであり、ARM命令の約80%をカバーします。メモリ制約の厳しいシステムではThumb命令でプログラムを記述すれば、性能をあまり落とさずにコード・サイズを圧縮できます。また、アーキテクチャによってはJavaのバイト・コードも実行できます(Jazelle機能)。これも命令セットと見ればバイト・コードなので、8ビット命令セットになります。データ処理単位としては32ビットです。

さらに、v7アーキテクチャ導入時にThumb-2が登場しました。Thumb-2命令セットでは16ビット長と32ビット長の命令が混在し、性能とコード・サイズのバランスをとっています。そのため、性能的にはARM命令セットに匹敵するものの、コード・サイズ

はThumb命令セットと同等程度という理想的な命令セットになっています。またThumb-2命令はThumb命令のスーパーサブセットであり、従来のThumb命令も実行可能です。

さらにJITコンパイラなどでの実行環境を実現するのに最適な命令セットであるThumb-2EE命令セットを搭載しているアーキテクチャもあります(表1)。

● データ長

ARMプロセッサは32ビット・プロセッサなので、ワード長は32ビット、ハーフ・ワードは16ビット、バイトは8ビットです。データ長としてはダブル・ワード(64ビット)も扱えますが、アーキテクチャのバージョンに依存します。

● エンディアン

ARMは、デフォルトではリトル・エンディアンですが、プロセッサ・コアによってはハードウェア的にビッグ・エンディアンに切り替えられます。図1にリトル・エンディアンにおけるデータのレジスタ・ロードのバイト順を示します。

● アドレス空間

現在のARMプロセッサのアドレス空間は4Gバイトです。初期のものの中には26ビット・アドレスもありましたが、現在はすべて32ビットです。また、I/O空間のような別アドレス空間は存在しません。したがって、I/Oを接続する際は、メモリ空間の一部を

表1 命令セット種類

実行できる命令の種類	
ARM命令(32ビット)	
Thumb命令	Thumb(16ビット, v4~v6)
	Thumb-2/2EE(16/32ビット, v7)
Javaバイト・コード(8ビット)	

ARM 命令セットの詳細

小林 達也

ARM プロセッサでは ARM アーキテクチャ標準の ARM 命令のほかに、コード・サイズを抑えられる Thumb 命令、Java プログラムを高速に実行できる命令、マルチメディア関連の命令などさまざまな命令が拡張されている。本章では ARM 基本命令から、各種拡張命令について解説する。 (編集部)

ARM プロセッサはシンプル、かつ省電力であることに重点をおいて設計されています。第1章で述べたように、ARM プロセッサはソフトウェアとハードウェアのバランスを最適化するために大胆な取捨選択をしました。その結果、ソフトウェアの視点から見たときに、ほかのプロセッサにはない独特の特徴があります。これらの特徴とともに、ARM プロセッサの命令セットについて解説します。

1 ARM 命令セットの特徴

すでに第1章で説明したように、ARM プロセッサは RISC アーキテクチャをベースに設計されたものです。このため、ARM プロセッサはほかの RISC プロセッサと同じように、いくつかの RISC としての特徴をもっています。その一方で、命令セットの構造にはほかの RISC プロセッサには見られないユニークな特徴があります。

● 演算と算術/論理シフトの1命令同時実行

ほとんどの演算命令で基本演算と同時に一方のオペランドに対する算術、または論理シフト演算を実行できます。わかりやすくいえば、算術/論理シフト命令と演算命令を同時に実行できるということです。たとえば、ARM プロセッサではレジスタ R0 にレジスタ R1 を8倍(3ビット左シフト)して加算し、結果を R0 へ代入する動作を、

```
ADD R0, R0, R1, LSL #3
```

と表記して1命令で行うことができます。これを応用すると、レジスタの3倍算や5倍算といった演算も1命令で実行できます。

● 多彩なアドレッシング・モード

ARM プロセッサは命令セットに応じて複数のアドレッシング・モードを持ちます。特にメモリ転送関係の命令では、前述のシフト命令と演算命令の同時実行を活用した非常に多彩なアドレス指定が可能です。これらのアドレッシング・モードは一見複雑ですが、ARM プロセッサ独特のハードウェア構成を無駄なく活用して効率の良い実行を可能にしています。

● 演算命令のフラグ・セット選択

多くの演算命令において、演算結果をフラグに反映する/しないを選択できます。

たとえば減算命令である SUB 命令で、R0 と R1 の各レジスタに同じ値が入っており、

```
SUB R0, R1, R0
```

としたとき、演算結果が0であることを示す Z フラグは変化しませんが、フラグ・セットを意味する記述子 'S' を SUB 命令に付けて、

```
SUBS R0, R1, R0
```

とした場合はフラグ・レジスタが更新され、Z フラグがセットされます。これを、次に説明する条件実行と組み合わせれば有効に活用できます。

● 命令の条件実行

ほとんどの命令で、フラグに応じて命令の実行と非実行(NOP)を決定できます。

たとえばレジスタ R0 が0以上であれば '1'、-1以下のときは '0' をレジスタ R1 に代入するプログラムを考えます。一般的なプロセッサでは比較命令と分岐命令を使用しなければならないため、リスト 1(a)のようなプログラムになります。これに対して ARM プロセッサでは、命令の条件実行の使用によりリスト 1(b)のように3命令で、しかも分岐命令を使用せず

仮想記憶だけじゃない。
組み込みシステムのセキュリティ強化に応用できる

MMUのメモリ保護機能を使ったプログラミング

中森 章

近年の組み込み機器の複雑化に伴い、MMU機能を使った大規模システムが作られるようになった。MMUを使えば、プロセスごとに独立した仮想アドレス空間を持つことができる上、一つのプロセスが暴走してもほかのプロセスやOSカーネルに影響を与えないという利点がある。また、ページ・スワップを使えば、少ないメモリを有効に使うこともできる。

本章では、OSの製作や、OSのない環境でメモリ保護機能を使うのに必要なMMUについて、ARM926コアを対象として解説を行う。(編集部)

組み込みプログラムが高度になり、また複雑になるにつれ、Symbian OS、Windows CE、LinuxといったOperating System(OS)の必要性が高まっています。特に出来合いのアプリケーション・プログラムは何らかのOSの下で動作することを前提としているのが普通です。MMU(Memory Management Unit)はOSを動作させるために必須の機能といわれています。私達が一般的に入手可能な評価ボードは汎用的な使用方法が想定され、OSが載ることを前提としており、そこに搭載されるプロセッサもMMU内蔵のものが使われます。

一方、MPU(Memory Protection Unit)機能を持つプロセッサは、OSを必要としない特定用途向けのSystem On a Chip(SoC)やApplication Specified Integrated Circuit(ASIC)に多く内蔵されます。しかし、私達がそのようなSoCやASICをプログラムする機会は少ないと思われるので、MPUの説明は今回は割愛します。

以降、私達が最も目にする機会が多いと思われるARM926コアのMMUについて説明します。プロセッサ・コアをARM926に特定していますが、ほかのARM9系のプロセッサ(MMUを持つもの)に対しては同じような方法でMMUを操作できます^{注1}。

1 MMUとは何か

本書の読者にMMUの解説をするのは釈迦に説法かもしれませんが、MMUの機能について簡単に説明

しておきましょう。

MMUには大きく分けて、

- アドレス変換
- メモリ保護

の二つの機能があります。

● アドレス変換—仮想アドレスを物理アドレスに変換する

これは仮想アドレスを物理アドレスに変換する機能です。プログラマに見えるアドレスが仮想アドレスです。もちろん、プロセッサも仮想アドレスに基づいて動作します。一方、メモリ側から見える本当のアドレスが物理アドレスです。

アプリケーション・プログラムのプログラマは、通常はプログラムがメモリのどこに存在するかを意識することなく、すべてのプログラムは同一のアドレス(たとえば0番地)から始まっているものとしてプログラミングします。「アプリケーションAは0x1000番地から始まり、アプリケーションBは0x2000番地から始まる」などと考えてプログラミングすることは、まずありません。すべて0番地から始まるという仮定でプログラミングします。逆にいうと、アプリケーションの数だけ同一の仮想アドレスが存在します。

しかし物理アドレスの場合は、メモリというデバイスのアドレスは一意なので、同一のアドレス上では一

注1：とはいえ、ARM920TやARM922TのTRM(Technical Reference Manual)にはMMUがARMv4準拠となっているのが気にかかる。ARM926のTRMでは、MMUはARMv5準拠となっている。

MMUが使えるようになったら次はキャッシュを制御してみよう

CPU性能を引き出す キャッシュ制御プログラミング

中森 章

ARM以外のCPUアーキテクチャでは、キャッシュ・コントローラのイネーブル・ビットをONにするだけで、キャッシュが有効になるものが多い。しかしARMアーキテクチャでは、そのような簡単な操作ではキャッシュは有効にならない。ここではCPU性能を引き出すための命令およびデータ・キャッシュの制御方法について詳しく解説する。
(編集部)

1 キャッシュ制御の基本

第3章ではMMUについて詳しく解説しました。またARMの場合、キャッシュを使うにはMMUをONにしなければならないことを示しました。通常の組み込み用途ではMMUを使う機会は少ないと思われます。しかし、キャッシュは、性能向上のためには必須です。ただし、データ・キャッシュを扱うためにMMUをONにしなければならないとすると、ユーザにとってはかなり苦痛と思われます。

しかしARMにおけるキャッシュの制御は、最初か

ら敬遠するような複雑なものではありません。本章では、仮想アドレスと物理アドレスが等しい条件の下でキャッシュの使用方法に関して説明したいと思います。

● キャッシュの概要

まずは、MMUの使い方の復習です。図1を見てください。仮想アドレス(修正仮想アドレス)は、TTBRレジスタによって指し示されるメモリ上のページ・テーブル(またはセクション・テーブル)を参照することによって、物理アドレスに変換されます。ここでは、アドレス変換自体はセクション・テーブルを使用した1レベルの変換方式を仮定しています。この場合、セクション記述子のCビットとBビットがキャッシュ

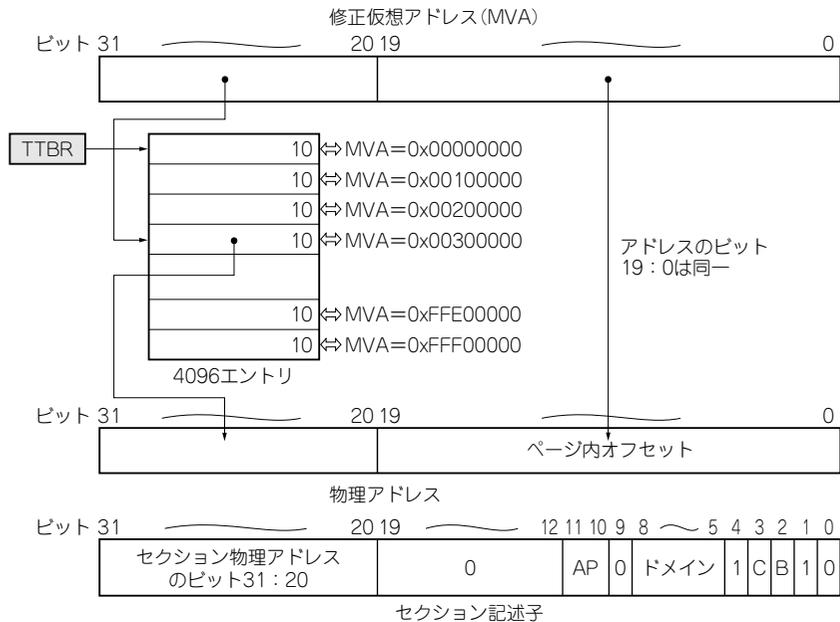


図1
セクションを使った
アドレス変換

第5章

PC/104バスによる拡張も容易な組み込み向けボード Armadillo-9

Cirrus Logic EP9315 搭載
評価ボードの概要と活用方法

花田 政弘 / 実吉 智裕

本章では、Cirrus Logic 社製 ARM9 コア CPU 内蔵 EP9315 を搭載した Armadillo-9 を紹介する。このボードには PC/104 バスが搭載されているので、各種 PC/104 対応拡張ボードとの組み合わせで必要なシステムを構築することもできる。またハードウェアの紹介の後、Linux 採用応用事例として MP3 再生システムを構築する。
(編集部)

● ARM9 コア搭載高性能 CPU ボード

「Armadillo-9 (アルマジロ 9)」は ARM9 コアを用いた Cirrus Logic 社製のプロセッサ EP9315 を搭載した、Linux 対応のシングル・ボード・コンピュータです。ARM7 コアを搭載した先代「Armadillo」が発表されてからすでに3年経過しました(執筆当時)。当時は組み込み機器の OS にも Linux を採用するという動きが表に見え始め、世界中で多くの組み込み Linux 企業が誕生しました。

携帯機器では圧倒的に強い ARM プロセッサですが、汎用プロセッサとして流通している種類は少なく、手軽に調達することのできる ARM プロセッサは、今でも多くはありません。そのため ARM を採用した汎用ボードも少なく、比較的低価格で Linux の動く状態で出荷されている Armadillo は、ある意味貴重な存在だったのかもしれません。

あれから3年が過ぎようとする今、ようやく上位機種種の「Armadillo-9」が登場しました。“-9”は ARM9 の9です。この間に「組み込み Linux」はすっかり定着し、Armadillo 以外にも多くの Linux 対応の汎用ボードが発表されました。

● 組み込みで Linux を採用するメリット

筆者の考える Linux を採用するメリットは、何といてもソフトウェア資産が多いことです。Linux カーネルを持つ多くの機能はもちろん、デバイス・ドライバからプロトコル・スタック、そしてアプリケーションまであらゆるレイアにおいてソフトウェアの資産を持っています。

Linux が動作しているからには Armadillo も Armadillo-9 もソフトウェアとしては大きな違いはあ

りません。強いて挙げれば当時のカーネルのバージョンは 2.4.16 でしたが、今回は 2.4.27 (執筆段階)になったことくらいです。しかし、さまざまなハードウェアの機能追加に伴い、そのハードウェアに対応したソフトウェアを追加しました。

ここではまず、追加されたさまざまなハードウェアの機能について解説し、後半ではソフトウェアの開発手順、そしてソフトウェアの実装例を紹介します。

1 Armadillo-9 の
ハードウェア構成

● 初代 Armadillo への要望

初代 Armadillo に対して要望の多かった機能としては、次のようなものがあります。

- (1) USB に対応すること
- (2) 何らかの形で画面出力機能を持つこと
- (3) コンパクト・フラッシュの I/O カードに対応させること

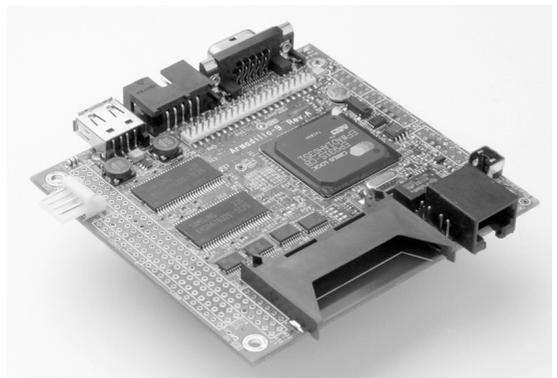


写真1 Armadillo-9(アットマークテクノ)の外観

Cirrus Logic EP9307 の メモリ・コントローラとブート手順

実吉 智裕

EP9307 は 200MHz のクロック周波数で動作する ARM920T コアを搭載したプロセッサである。キャッシュ以外にはメモリを内蔵していないが、Ethernet コントローラや USB ホスト・コントローラ、LCD コントローラなど、多数の周辺機能を備えている。ここでは EP9307 を採用した CPU ボードを設計する上で理解が必要な電源やリセット周辺、およびメモリ・コントローラや各種ブート手順について解説する。(編集部)

1 EP93xx シリーズと EP9307

● EP93xx シリーズ概要

EP93xx シリーズは CPU コアに ARM920T を採用した米国 Cirrus Logic 社の SoC(System on a Chip) です。低価格の EP9301 から、グラフィックス・エンジンや PCMCIA コントローラを内蔵した EP9315 まで、五つの品種があります。すべての製品において共通しているのは、メモリ・コントローラや割り込みコントローラなどのプロセッサとして基本的な部分と、100 Mbps 対応の Ethernet MAC および USB ホスト・コントローラ(フル・スピードの 12Mbps)を搭載していることです。

各製品の機能的な違いを表 1 に示します。

● EP9307 概要

本章では EP9307 を中心に必ず使用される機能について解説し、その EP9307 を採用した事例として Linux の動作する小型の EP9307 ボード「Armadillo-200 シリーズ」を紹介します。EP9307 はシリーズの中でも最も小さいパッケージ(14mm 角の 272 ピン TFBGA)を採用しながらも豊富な周辺回路を持っています。200

MHz という高い処理能力と 750mW 以下の低消費電力を実現し、シンクライアントやネットワーク・オーディオ機器、計測器、POS 端末のような、画面とネットワーク、USB を必要とするアプリケーションに向いています。EP9307 の機能ブロック図を図 1 に、機能の概要を表 2 に示します。

EP9307 のプロセッサ・コアには ARM920T を採用しています。ARM920T は ARM9TDMI をベースに、16K バイトの命令キャッシュ、16K バイトのデータ・キャッシュ、そして MMU (Memory Management Unit) が搭載されています。

ARM9TDMI は ARMv4T のアーキテクチャをとる整数演算コアで、Thumb 命令に対応しています。ARM920T には浮動小数点演算命令は搭載されていませんが、EP93xx シリーズ(EP9301 を除く)には Maverick Crunch と呼ばれる浮動小数点演算や 64 ビット演算などに対応したコプロセッサが搭載されています。

2 EP9307 の電源および クロック周辺

それでは、EP9307 を使うに当たって理解が必要な部分、特に電源や外部バス周辺、およびブート手順や

表 1 EP93xx シリーズ一覧

型番	クロック [MHz]	LAN	PCMCIA	IDE	USB ホスト	LCD コン トローラ	グラフィック ス・エンジン	浮動小数点 演算ユニット	タッチ・パネル/ A-D コンバータ	パッケージ
EP9301	166	○	—	—	2	—	—	—	5 ADC	208 TQFP
EP9302	200	○	—	—	2	—	—	○	5 ADC	208 TQFP
EP9307	200	○	—	—	3	○	○	○	8 ワイヤ	272 TFBGA
EP9312	200	○	—	2	3	○	—	○	8 ワイヤ	352 PBGA
EP9315	200	○	○	2	3	○	○	○	8 ワイヤ	352 PBGA

Digi International NS9360 と 評価ボード AZ9360MB の概要

長岡 美紀雄 / 松本 哲明 / 小澤 智幸 / 奥村 徹也

ここでは、ARM9 ベースの CPU である NS9360 を搭載した CPU ボード AZ9360MB について解説する。CPU ボードは、パワー ON 後にシステム・クロックの設定や SDRAM コントローラ、割り込みコントローラなどを適切に設定しなければ動作しない。また、ARM9 系 CPU はパワー ON 後に 0 番地から実行を開始するが、このアドレスに SDRAM が割り当てられていることがある。このような場合の対処方法など、CPU ボードを使うための手法について解説する。 (編集部)

1 小型 CPU ボード AZ9360MB の概要

AZ9360MB は、米国 Digi International 社 (旧 Net Silicon 社) 製の ARM プロセッサ「NS9360」を中心に、フラッシュ ROM 8M バイト、SDRAM 32M バイト、

オン・ボード周辺機器として Ethernet, USB ホスト/ターゲット機能, シリアル, JTAGなどを搭載した小型 CPU ボードです。仕様を表 1 に、ブロック構成を図 1 に、外観写真を写真 1 に示します。本 CPU ボードは小型サイズながら (名刺 2 枚分)、豊富な周辺機能を搭載しているので、ARM9 システムの試作や実験、

表 1
AZ9360MB の仕様

機能	仕様
CPU	NS9360 (米国 Digi International 社)
動作周波数	内部 177MHz 外部 88.5MHz
ROM	フラッシュ ROM 8M バイト (16 ビット幅)
RAM	SDRAM 32M バイト (32 ビット幅)
USB	ホスト機能 × 2 チャンネル USB フル・スピード (12Mbps) / ロー・スピード (1.5Mbps) 各チャンネル +5V, 最大 500mA 供給 ターゲット機能 × 1 チャンネル セルフ・パワー動作
Ethernet	1 チャンネル, 10Base-T/100Base-TX
UART	2 チャンネル (RS-232-C レベル) フロー制御ありポート (シリアル・ポート A) フロー制御なしポート (シリアル・ポート C)
リアルタイム・クロック (RTC)	NS9360 内部機能による (バックアップ機能なし) RTC デバイス (セイコーエプソンの RX-8025SA) による (バックアップ電源は外部供給) ^注
MMC/SD カード	NS9360 内蔵機能による。SPI インターフェース ^注
CPU JTAG	JTAG ICE 用ポート
汎用ポート	DIP スイッチ 5 ポート 汎用 LED 4 ポート
拡張インターフェース	32 ビット幅 (88.5MHz 動作)
入力電源	+5V ± 5%
内部生成電源	+3.3V/+1.5V
消費電力	最大 8W (ボード単体: 3W, USB 電源供給: 5W)
ボードの外寸寸法	90mm × 110mm (突起物含まず)

注: オプション機能

Freescale i.MX25 シリーズ概要 と Armadillo-400 の使い方

波塚 朋広

本章では Freescale Semiconductor 社製 ARM9 コア搭載 i.MX25 シリーズについて解説する。まず i.MX25 シリーズの概要について紹介した後、i.MX25 を採用した CPU ボードを設計するときに必要な電源やリセット、クロック、メモリ・コントローラ周辺について解説する。最後に i.MX257 搭載評価ボード Armadillo-400 シリーズについて紹介する。

(編集部)

1 i.MX25 シリーズの概要

● i.MX25 シリーズの用途

i.MX25 シリーズは、CPU コアに ARM926EJ-S を採用した米国 Freescale Semiconductor 社製のマルチメディア・アプリケーション・プロセッサです。ポータブルなマルチメディア製品を対象とした多くの周辺機

能を搭載し、優れた電力管理テクノロジーを採用しているのが特徴です。次のようなマルチメディア機器やヒューマン・インターフェース・アプリケーションを開発するのに適しています。

- ポータブル・メディア・プレーヤ
- デジタル・サイネージ
- デジタル・フォトフレーム
- ヒューマン・マシン・インターフェース (HMI)

表1 i.MX25 シリーズの周辺機能の比較

	i.MX251	i.MX253	i.MX255	i.MX257	i.MX258
コア	ARM926EJ-S				
CPU 速度	400MHz				
パラレル ATA	-	○	○	○	○
LCD コントローラ	-	○	○	○	○
タッチ・スクリーン・コントローラ	-	-	○	○	○
カメラ・インターフェース	-	-	○	○	○
CAN	○	-	○	○	○
ESAI	○	-	○	○	○
SIM	-	-	○	○	○
セキュリティ	-	-	○	-	○
Ethernet	○	○	○	○	○
USB	○	○	○	○	○
A-D コンバータ	○	○	○	○	○
SD/SDIO/MMC	○	○	○	○	○
UART, I ² C, SPI, SSI	○	○	○	○	○
AEC-Q100 認証	○	-	○	-	-
動作温度	-40 ~ 85 °C	-20 ~ 70 °C, -40 ~ 85 °C	-40 ~ 85 °C	-20 ~ 70 °C, -40 ~ 85 °C	-40 ~ 85 °C
パッケージ	MAPBGA (400ピン) 17 × 17mm/ 0.8mm ピッチ	MAPBGA (400ピン) 17 × 17mm/ 0.8mm ピッチ	MAPBGA (400ピン) 17 × 17mm/ 0.8mm ピッチ	MAPBGA (400ピン) 17 × 17mm/0.8mm ピッチ, MAPBGA (347ピン) 12 × 12mm/0.5mm ピッチ	MAPBGA (400ピン) 17 × 17mm/ 0.8mm ピッチ
用途	車載	民生, 産業	車載	民生, 産業	産業

リアルタイム・ビデオ・エンコーディング機器開発を例にした

Freescale i.MX27 採用 画像処理システムの開発事例

井倉 将実

Freescale Semiconductor 社の i.MX27 は、高機能携帯機器向けでアプリケーション・プロセッサと呼ばれている。ここでは i.MX27 を採用したシステムを設計するときに必要な、クロックやリセット、複数系統必要な電源周辺や I/O バンクの電圧設定などについて解説する。
(編集部)

● i.MX27 について

Freescale Semiconductor 社が提供する i.MX27 は、ARM926EJ-S コアを中核に、携帯情報端末などに最適な機能をワンチップに統合した CPU です。i.MX27 は、アプリケーション・プロセッサという名称が付けられているとおり、この CPU 一つで、高機能携帯端末を作るにはじゅうぶんすぎるほどの、非常に多くの機能が搭載されています。特に画像処理関連のエンジンを内蔵している点がポイントです。

ここでは i.MX27 を採用した製品開発を行う場合に、製品マニュアルでは見落としがちなハードウェア設計における注意点について解説します。

● 開発した機器について

ここで紹介する i.MX27 を搭載した画像処理システムは、Freescale Semiconductor 社が提供する i.MX27-BSP と呼ばれるリファレンス・ボードを参考に、外部カメラからの映像を入力してソフトウェアで H.264 方式に変換し、SD カードまたは Ethernet を介して映像情報を保存するものです。さらにオンボードに搭載された LCD パネルで画像のプレビューを見ることがもできます。

図 1 に i.MX27 内部ブロックを示します。このうちの網掛け部分が、ここで設計したシステムで使用している機能です。図 2 に設計したシステムのブロック図を示します。

今回開発したシステムで使用した機能の多くが、i.MX27-BSP に付属の回路図で紹介されているため、設計そのものにそれほど難儀はないかもしれません。

● ドキュメントが少ない

しかしながら、i.MX27-BSP の回路図やマニュアルには「なぜ、この端子を使うのか」、「なぜ、この機能

をこのように使うのか」はまったく説明されていません。ほとんどの機能は端子をつなげれば動くというレベルではなく、i.MX27-BSP に付属しているソース・コードを追いかけて機能を解析しながら使用した経緯があります。

アプリケーション・プロセッサとはいえ、単体で電源や LCD パネル、キーボード入力、通信、ストレージ機能などをすべて行うことはできないため、外部に ASSP (特定用途の LSI) や専用 LSI を搭載して使用することになります。しかしこのときに、BSP に搭載されているデバイスとは異なるものを使った瞬間から、あの機能はどう使うのか？この端子はどこに接続すればよいのか？などと、機能が豊富なだけに理解に苦しむものもあります。

このような場面に陥った場合は、ハードウェア仕様書だけでなく、i.MX27-BSP に添付された Linux のデバイス・ドライバやソース・コード類にも目を通すことを推奨します。

特に、i.MX27 が注目される H.264/MPEG-4 など向けの映像エンコーディング支援機能とビデオ出力機能については、マニュアルの記述もほとんどなく、ソフトウェアを実際に動かしてみることで動作が理解できるでしょう。

1 クロックとリセット系統の設計

● i.MX27 のクロック系

組み込み機器用に開発された i.MX27 は、すでに説明したように非常に多くの機能を有しているため、さまざまな内蔵機能(モジュールと称する)が ARM926 コアとは異なるクロックで動作します。このため i.MX

LCD コントローラやカメラ・モジュールもつなげられる

Atmel AT91SAM9 シリーズと AT91SAM9260 の使い方

大野 俊治

本章では、Atmel 社製 ARM9 コア搭載 AT91SAM9 シリーズについて解説する。まず SAM9 シリーズの概要と、SAM7 シリーズと比較して内蔵周辺機能の違いについて解説する。そして SAM9 シリーズで特徴的な MCI と ISI インターフェースについて詳しく解説し、最後に応用事例としてカメラ取り込み & SD カードへの保存するシステムについて解説する。

(編集部)

1 AT91SAM9 シリーズ概要

● ARM926EJ-S コア内蔵シリーズ

AT91SAM9 シリーズは、ARM926EJ-S コアを搭載した米国 Atmel 社の ARM9 プロセッサです。周辺機能の多くは同社の ARM7 マイコンである AT91SAM7 シリーズとほぼ同じとなっており、同シリーズで開発したアプリケーションを容易に移植することが可能です。

SAM9 シリーズはすべての CPU が外部メモリとのインターフェース (EBI) を持っており、多くの CPU では SAM7 シリーズではサポートされていなかった USB ホストやイメージ・センサ入力、LCD 出力などの機能がサポートされていることが特徴です。ここでは、AT91SAM7 シリーズとの違いを中心に、AT91SAM9 シリーズの持つ機能を説明していくこととします。

ます。

● シリーズ概要

表 1 に SAM9 シリーズの主要 CPU 一覧を示します。SAM7 シリーズがフラッシュ ROM を内蔵していたのに対し、SAM9 シリーズでは AT91SAM9XE を除いて内蔵フラッシュ ROM を持っておらず、外部デバイスからブートすることを基本としています。

(1) AT91SAM926x シリーズ

AT91SAM9260 は、SAM9 シリーズのうちで最も基本的な構成を持つ CPU です。Ethernet と CMOS カメラを接続できるイメージ・センサ・インターフェース (ISI) を持ち、ネットワーク・カメラのような用途に使用することができます。

AT91SAM9261 は、Ethernet と ISI の代わりに LCD インターフェースを持っており、PDA のような端末用途に適しています。

AT91SAM9263 は、Ethernet と ISI、LCD インター

表 1 SAM9 シリーズの CPU 一覧

デバイス名	プロセッサ・クロック (Hz)	命令/データ・キャッシュ (バイト)	SRAM (バイト)	EMAC	ISI	LCD	USB ホスト	パッケージ
AT91SAM9260	190M	8K/8K	4K × 2	○	○	×	○	LFBGA 217, PQFP 208
AT91SAM9261	190M	16K/16K	160K	×	×	○	○	LFBGA 217
AT91SAM9263	190M	16K/16K	96K	○	○	○	○	TFBGA 324
AT91SAM9G10	266M	16K/16K	16K	×	×	○	○	LFBGA 217
AT91SAM9G20	400M	32K/32K	16K × 2	○	○	×	○	LFBGA 217
AT91SAM9G45	400M	32K/32K	64K	○	○	○	○	TFBGA 324
AT91SAM9M10	400M	32K/32K	64K	○	○	○	○	TFBGA 324
AT91SAM9R64	240M	4K/4K	64K	×	×	×	×	LFBGA 144
AT91SAM9RL64	240M	4K/4K	64K	×	×	○	×	LFBGA 217
AT91SAM9XExxx	190M	8K/16K	16K, 32K	○	○	×	○	LFBGA 217, PQFP 208

ARM926EJ-S コア+フラッシュROM & RAM内蔵1チップ・マイコン

Atmel AT91SAM9XE シリーズ搭載 評価ボードの設計と使い方

山武 一朗

ARM9 クラスの CPU は、メモリを外付けして使うのが一般的である。しかしここで紹介する Atmel 社製 AT91SAM9XE シリーズは、フラッシュROM と SRAM を内蔵し、1 チップ・マイコンとしても使える。数百 K バイトの ROM と数十 K バイトの RAM で動作する ITRON などの小規模 OS を採用する場合は、1 チップでシステムを構成できる。

(編集部)

1 ARM9 CPU カードの ハードウェア

ここでは筆者が設計した ARM9 搭載 CPU カードの設計事例と使い方について解説します。この CPU カードとはあるマザーボードに搭載可能なオプション CPU カードとして設計しましたが、CPU カード上にローカル・メモリを搭載していることもあり、この CPU 単独でも Linux を起動させられるだけの能力を持ちます。マザーボード側については本章の範疇ではないので、解説は省略します。

この CPU カードは、最大クロック周波数 200MHz 動作の ARM9 コアを内蔵した AT91SAM9XE256 (米国 Atmel 社) を搭載しています。CPU カードの外観を写真 1 に、仕様を表 1 に示します。LAN と USB ホスト、SD/MMC カード・ソケットは CPU カード本

体とは別に子基板で拡張する形としました。

● AT91SAM9XE256 の特徴

まず搭載している CPU の特徴について説明します。Atmel 社は ARM7 や ARM9 系の CPU コアを内蔵したマイコン・ファミリとして「AT91SAM シリーズ」を展開しています。型名の SAM の後の数字が 7 なら ARM7、9 なら ARM9 のコアを内蔵しています。AT91SAM9XE シリーズは名前からわかるように ARM9 コアを内蔵しています。ARM9 コアにもいくつか種

表 1 ARM9 (AT91SAM9XE) CPU カードの仕様

搭載CPU	AT91SAM9XE256
CPU コア	ARM926EJ-S
最大動作クロック	CPU コア：約 198MHz
CPU 内蔵周辺	汎用 I/O、A-D コンバータ、高性能タイマ、各種シリアル通信コントローラ、USB ホスト/ターゲット、Ethernet (論理層のみ)、SD/MMC カード・コントローラほか
フラッシュROM	16 ビット幅、8M バイト
SDRAM	32 ビット幅、64M バイト
マザーボード接続バス	32 ビット幅ローカル・バス
各種 インターフェース	デバッグ・インターフェース (ARM 標準 20 ピン JTAG) CPU 内蔵 USB ターゲット (標準 B コネクタ) CPU 内蔵シリアル・ポート 1 (DBGU または USART0 の排他使用) CPU 内蔵シリアル・ポート 2 (USART1) 拡張インターフェース (USB ホスト、Ethernet 用 MII、SD/MMC カード)
CPU カード・ オンボード機能	LED：4 ビット、ディップ・スイッチ：4 ビット、タクト・スイッチ：1 個 (割り込み発生可能)
供給電源	3.3V、最大 3A (マザーボード側から供給)

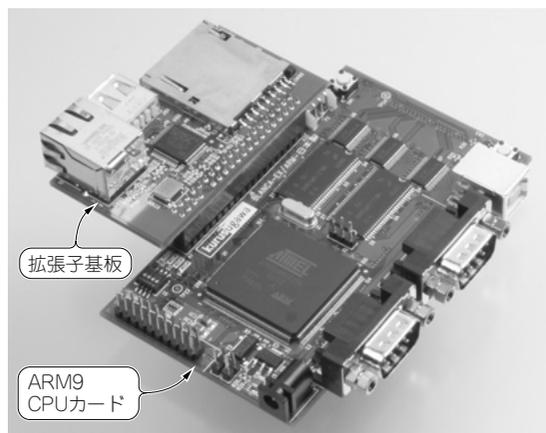


写真 1 ARM9 (AT91SAM9XE) CPU カードの外観

シリアル・コントローラや GPIO を直接制御してみよう

Marvell ARM プロセッサ搭載 玄箱 PRO の活用事例

光永 法明

玄箱 PRO という Linux ボックス開発キットがある。ブート・ローダに U-Boot, OS として Linux というオープン・ソースのソフトウェアを使い、自由にカスタマイズして使うことができる。この玄箱 PRO には ARM 系プロセッサが搭載されている。この CPU を骨まで使いこなすための第一歩として、この CPU 内蔵のシリアル・コントローラおよび汎用 I/O を直接制御するプログラムを作成してみる。(編集部)

1 玄箱 PRO と U-Boot

● 玄箱 PRO とは

玄箱(クロバコ)PRO は、玄人志向の Linux ボックス開発用の組み立てキットです(写真 1)。シリアル ATA(SATA)ハード・ディスク(HDD)を購入して取り付け、簡単なセットアップをすれば Linux が動作します。表 1 に示すように CPU に ARM9 ベースの MV88F5182 を採用しています。本章では玄箱 PRO に内蔵された U-Boot を使い、OS 起動前に直接レジスタを操作して動作を確かめる方法を紹介します。

● Das U-Boot とは

Das U-Boot^{注1}は ARM, MIPS, PowerPC など複数のアーキテクチャに対応した組み込みシステム向けの

ブート・ローダです。CPU リセット直後に、メモリ周りや周辺デバイス(ディスクや Ethernet)を初期化し、ディスクやネットワークから OS を起動することができます。ライセンスは GPL のバージョン 2 です。U-Boot には、ROM モニタのような機能があり、メモリのダンプや書き換え、バイナリを読み、実行することができます。

● 玄箱 PRO の U-Boot

玄箱 PRO でも U-Boot が採用されており、シリアル・コンソールから操作できます。図 1 に玄箱 PRO に搭載されている U-Boot のコマンドの一覧を示します。SATA インターフェースの初期化に使う ide コマンド、リセットする reset コマンド、ext2fs 上のファ

表 1 玄箱 PRO の主な仕様

CPU	ARM9 互換 Marvell 製 88F5182 400MHz
メモリ	DDR2-SDRAM 128M バイト
ドライブ・ベイ	3.5 型/2.5 型 SATA HDD 用 × 1
BootROM	256K バイト・フラッシュ ROM (NOR 型)
System 領域 +Data 領域	256M バイト・フラッシュ ROM (NAND 型)
外部インター フェース/ コネクタ	USB TypeA × 2 RJ-45 × 1(1000Base-T/100Base-TX/ 10Base-T 対応)
内部コネクタ	PCI-ExpressX1 × 1(ボード取り付け スロット空間はない) SerialATA × 2(うち 1 個はドライ ブ・ベイ用)
内部スルー・ホール (ピン・ヘッダ用)	UART(コンソール) × 1, GPIO × 2 I ² C × 1, JTAG(ARM20 ピン) × 1
外形寸法	60(W) × 163.3(H) × 215.5(D)mm

写真 1
玄箱 PRO の外観注 1 : <http://www.denx.de/wiki/U-Boot/WebHome>

そのほかの ARM9 コア搭載 CPU 評価ボード

土居 敬治 / 芹井 滋喜

1 NXP LPC3131 搭載 評価キット

● LPC3131 とは

NXP Semiconductors(以下 NXP)社製の LPC3131 はオンチップ PHY 内蔵ハイ・スピード OTG/ホスト/デバイス(ターゲット)USB 2.0 インターフェース, NAND 型フラッシュ ROM, SDRAM などの外部メモリ・コントローラ, MMC/SD などのメモリ・カード・インターフェース, I²S, I²C, LCD インターフェースを備え, 最小の外部部品で最近のあらゆる組み込み機器のメイン・コントローラを構成できます。

図 1 に LPC3131 の内部ブロック図を示します。主な特徴は次のようになります。

- CPU
 - 180MHz 動作 ARM926EJ-S コア搭載
 - 16K バイトのデータ・キャッシュおよび命令キャッシュ
 - メモリ・マネジメント・ユニット(MMU)内蔵
- 内部メモリ
 - 192K バイト SRAM
- 外部メモリ・インターフェース
 - 8 ビット ECC 付き NAND 型フラッシュ ROM コントローラ
 - SDRAM, SRAM を接続できる 8/16 ビット・マルチポート・メモリ・コントローラ
- 通信および外部接続機能
 - オンチップ PHY 内蔵ハイ・スピード USB 2.0 (OTG, ホスト, デバイス)
 - I²S バス・インターフェース×2
 - マスタ/スレーブ SPI
 - マスタ/スレーブ I²C バス・インターフェース×2
 - 高速 UART
 - メモリ・カード(MMC/SD/SDIO/CE-ATA) インターフェース
 - 4 チャンネル 10 ビット A-D コンバータ
 - 4/8/16 ビット LCD インターフェース

● システム機能

- ダイナミック・クロック・ゲーティング/スケールリング
- 複数のパワー・ドメイン
- SPI 接続シリアル・フラッシュ, SD/MMC カード, UART, USB からのブートが可能
- DMA コントローラ
- 32 ビット・タイマ×4
- ウォッチドッグ・タイマ
- PWM モジュール
- 乱数発生器
- 汎用入出力ピン
- 柔軟で高機能な割り込み
- バウンダリ・スキャンおよび ARM デバッグ・アクセス用 JTAG

● 動作電圧と周囲温度

- コア電圧 1.2V
- 入出力電圧 1.8V/3.3V
- 周囲温度 - 40 ~ + 85 °C

● パッケージ

- TFBGA180 パッケージ
- 12 × 12mm, 0.8mm ピッチ

● LPC3131 評価キット

LPC3131 を搭載した LPC3131 評価キット (IAR システムズ)について紹介します(写真 1)。このキットには, 組み込みアプリケーションを設計, 開発, 実装, テストをするために必要なすべてのハードウェアおよびソフトウェアが同梱されています。

LPC3131 評価キットの内容は次のようになります。

- Embedded Artists 社製 LPC3131 (ARM9) 評価ボード
- JTAG ICE (IAR J-Link Lite)
- USB ケーブル
- AC アダプタ
- コンパイラ・デバッガ評価版 (IAR Embedded Workbench)
- ステートマシン設計ツール評価版

第13章

ARM11 コア搭載 CPU モジュールに
ビデオ/オーディオ/USBなどを拡張する

Freescale i.MX31 搭載 CPU モジュール対応拡張ボードの設計

波塚 朋広 / 実吉 智裕

組み込み機器開発向けの小型 CPU ボード Armadillo-500(開発：アットマークテクノ)を使った拡張ボード設計に必要な技術要素を解説する。Armadillo-500 は、Freescale Semiconductor 社製の i.MX31 (ARM11 コア搭載のアプリケーション・プロセッサ) を搭載し、組み込み機器開発のためのさまざまな工夫がなされている。信号マルチプレクスや外部メモリ・インターフェース、I/O コントローラ、GPIO などを使い、メモリや USB、オーディオなどの機能を拡張する方法を解説する。(編集部)

1 ARM11 コア搭載 CPU モジュール 「Armadillo-500」とは？

● Armadillo シリーズと Armadillo-500

Armadillo シリーズは、ARM プロセッサを採用した小型 CPU ボードです。CPU ボード上で動作する OS として Linux が提供され、Windows Embedded CE や μ ITRON も搭載できます。このシリーズは 2001 年に開発された初代 Armadillo (HT1070) に始まり、Armadillo-9、Armadillo-200 シリーズ、Armadillo-300 と発展してきました(表 1)。それぞれの ARM 世代を採用した Armadillo は、汎用 CPU ボードとして、なるべく多くの用途で利用しやすいように設計されています(少なくとも設計者はそのつもり)。

Armadillo-500 (以下、A500) は ARM11 を採用しており、Armadillo と名の付いた CPU ボードの中で最も拡張性の高いボードです(写真 1)。

● A500 のコンセプト

A500 は、「機能特化した組み込み機器のプラットフォーム」として開発されました。組み込みの世界では、

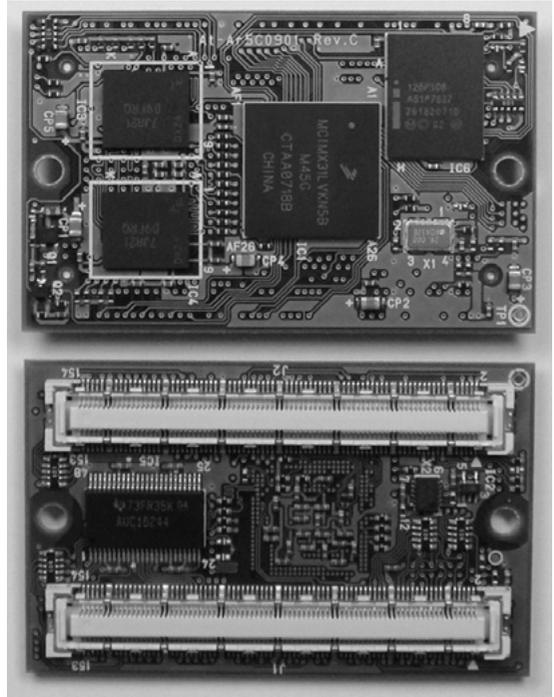


写真 1 Armadillo-500 の CPU モジュール

表 1 Armadillo シリーズの発表時期と搭載 CPU

製品名	発表時期	搭載 CPU のメーカーと型番	CPU コアの種類	CPU 動作クロック
Armadillo (HT1070)	2001 年 11 月	Cirrus Logic EP7312	ARM720T	74MHz
Armadillo-J	2003 年 10 月	Digi International NS7520	ARM7TDMI	55MHz
Armadillo-9	2004 年 7 月	Cirrus Logic EP9315	ARM920T	200MHz
Armadillo-210	2005 年 11 月	Cirrus Logic EP9307	ARM920T	200MHz
Armadillo-220/230/240	2006 年 4 月	Cirrus Logic EP9307	ARM920T	200MHz
Armadillo-300	2006 年 11 月	Digi International NS9750	ARM926EJ-S	200MHz
Armadillo-500	2007 年 5 月	Freescale i.MX31/i.MX31L	ARM1136JF-S	400MHz

高性能携帯端末で使われる XScale プロセッサ

Marvell PXA310 の特徴と 周辺機能の概要

保坂 一宏

ARM アーキテクチャは、ARM9、ARM10、ARM11 と高性能化してきたが、もう一つ独自に発展してきたアーキテクチャとして XScale がある。グラフィックス・アクセラレータ内蔵 LCD コントローラや SD コントローラ、USB など各種シリアル・コントローラなどを内蔵し、高性能な PDA(Personal Digital Assistant) でよく使われている CPU である。
(編集部)

1 PXA シリーズの歴史

PXA310 は、Marvell Technology Group Ltd.(以下 Marvell)社が提供する、第3世代 XScale マイクロアーキテクチャを採用したアプリケーション・プロセッサです。XScale は、ARM v5TE ベースのアーキテクチャで Intel 社により開発されました。

XScale の源流は、Digital Equipment Corporation (DEC)社が開発した StrongARM にあります。1997 年、DEC の半導体部門が Intel 社に売却され、後の XScale が誕生します。しかし Intel 社で誕生した XScale も、PXA シリーズは 2006 年に Marvell 社に売却されました。

ここでは、歴代の PXA シリーズのアプリケーション・プロセッサについての概要を説明しておきたいと思います。

● 第1世代 PXA25x/26x シリーズ

PXA250/PXA210 は 2002 年 2 月に発表された、PXA シリーズ最初のアプリケーション・プロセッサです。PXA250 は、最高動作周波数 200/300/400MHz の高性能、低消費電力モードと細かな電力制御、マルチメディア機能の処理能力向上、豊富な周辺 I/O などを特徴としています。PXA210 は、最高動作周波数が 200 MHz になっており当時の携帯電話やエンタリ・レベルの PDA 向けとなっています。

PXA255 は 2003 年 3 月に発表された PXA250 の後継となるプロセッサです。PXA250 のバグ・フィックスや機能アップを行ってきたもので、当初は PXA250 の C1 ステップ品となっていました、パフォーマンス

スがアップしたこともあり名称を PXA255 としてリリースされたものと考えられます。最高動作周波数、パッケージも PXA250 と同一です。機能としては、内部バスが 100MHz から 200MHz に高速化されたことと、ライト・バック・キャッシュに対応したこと、消費電力の低減が挙げられます。

PXA260/261/262/263 は PXA255 からパッケージを小型化し、さらにフラッシュ ROM を積層したシリーズです。

● 第2世代 PXA27x シリーズ

PXA270 は 2004 年 4 月に発表された第2世代の XScale アーキテクチャのアプリケーション・プロセッサです。最大動作周波数は 624MHz に高速化され、ワイヤレス MMX (パソコン系の MMX 命令と互換性が高いマルチメディア処理向け命令セット)、ワイヤレス・スピードステップによるクロックおよび電圧の可変制御、クイック・キャプチャ・テクノロジーによるカメラ対応が大きな特徴です。また、内蔵 SRAM や周辺 I/O 機能が強化されています。

PXA270 のリリースと同時にフラッシュ ROM を積層した PXA271/272/273 がラインナップされています。併せて、PXA27x シリーズと同時に使用するマルチメディア・アクセラレータ・チップ 2700G がリリースされました。2700G は PowerVR ベースのアクセラレータ・チップで VGA での MPEG-2 や MPEG-4、WMV9 などでの表示が可能となっています。また、SXVGA の解像度でのデュアル・ディスプレイに対応しています。

● 第3世代 PXA3xx シリーズ

PXA3xx シリーズは、2006 年 11 月に PXA シリーズ

初代 XScale のクロックや電源、メモリ、内蔵機能の概要

Marvell PXA25x/PXA26x アプリケーション・プロセッサ紹介

保坂 一宏

最近の高機能化した PDA では、XScale プロセッサが採用されている例が多い。ここでは、XScale プロセッサとはどんな CPU なのか、また PXA255 のアーキテクチャや内蔵機能など、ハードウェアの概要について解説する。(編集部)

PXA25x/PXA26x は、XScale マイクロアーキテクチャを採用したアプリケーション・プロセッサです。ハンドヘルド機器、ワイヤレス機器などの製品に適しています。PXA25x/26x シリーズは、数々の PDA (Personal Digital Assistants) などに実装された実績があります。

PXA25x/PXA26x に採用されている XScale マイクロアーキテクチャは、ARM アーキテクチャ V5TE に準拠しており、前身となる StrongARM とアプリケーション・コード・レベルで互換性を持っています。また、スーパー・パイプライン構造、40 ビット・アキュムレータ、SIMD などにより、マルチメディア系の処理も高速に実行が可能です。PXA25x/PXA26x ではさらに、各種 OS、ミドルウェア、ライブラリ、ソフト

ウェア/ハードウェアの開発環境が整備されています。

図 1 に PXA25x/PXA26x の内部構造を、表 1 に機能の一覧を示します。また PXA26x は Intel Strata Flash メモリがチップ内にスタックされて 1 チップとなり、さらにサイズも小さくなっています。表 2 に PXA26x のフラッシュ容量の違いを示します。

なお、すでに第 14 章で解説しているように、PXA 25x/26x は世代の古い XScale なので、新規にハードウェアを設計する場合は PXA310 以降の新しい世代の CPU を採用するのがよいでしょう。とはいえ、一時は PXA25x/26x を搭載した PDA などがかかり世の中に出回っていたので、本章と次章で解説してみたいと思います。

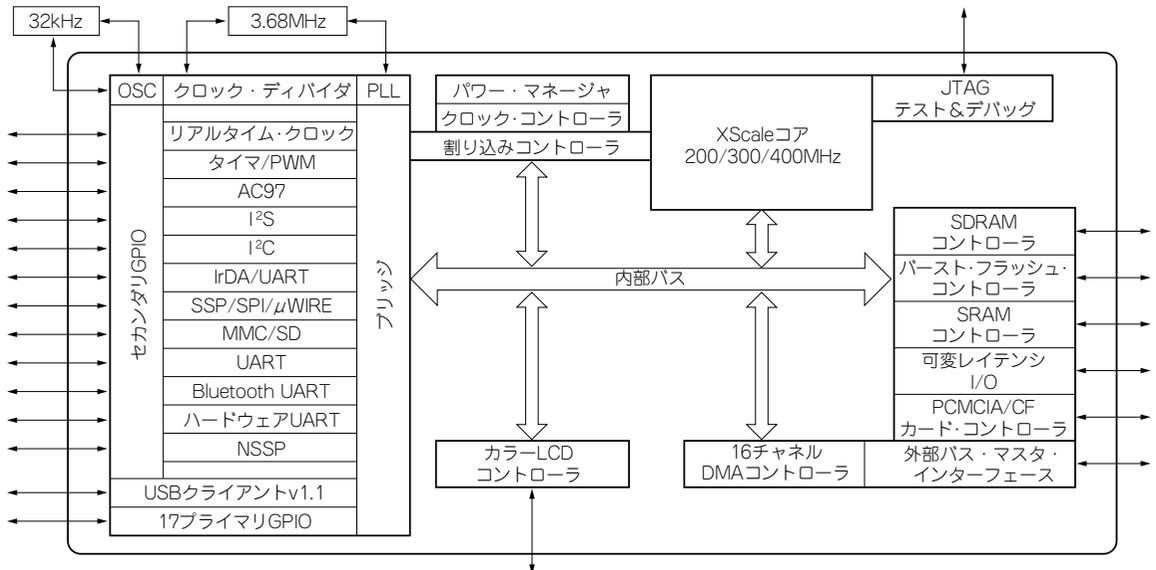


図 1 PXA255 回路ブロック

XScale プロセッサのプログラミング事例

保坂 一宏 / 山本 繁寿

本章では PXA250 を搭載した評価ボードを使って、タイマ割り込みプログラムやプッシュ・スイッチによる外部割り込みプログラム、シリアル通信プログラムなどを作成し、XScale シリーズの具体的なプログラミング事例について解説する。(編集部)

1 外部メモリの初期化例

● 評価ボードのメモリ・マップ

図 1 に今回使用した PXA250 搭載評価ボードのメモリ・マップを示します。PXA25x/26x シリーズを使う場合、CS0 にはフラッシュ ROM を、SDRAM0 には SDRAM を接続するのが一般的なもので、たいていの評価ボードは、図 1 のメモリ・マップとほとんど同じ割り当てになります。

CPU はリセット直後、CS0 の空間にアクセスしていくので、ここには ROM が必要になります。本評価ボードではフラッシュ ROM を接続しています。CS0 のバス幅などの設定は CPU の BOOT_SEL 信号ピンで行いますが、CPU ボードのフラッシュ ROM は ROM ソケットではなくフラット・パッケージを実装し、交換を想定していないので、本評価ボードの仕様として BOOT_SEL[2~0] は“000”で固定しています。

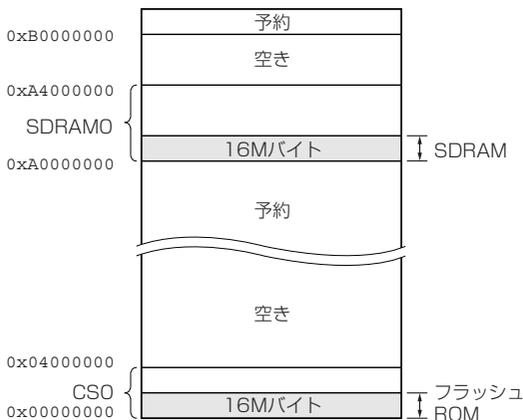


図 1 CPU ボードのメモリ・マップ

RAM は SDRAM バンク 0 に PC100, CL=3 の仕様の SDRAM を接続しています。

● メモリ・コンフィグレーション・レジスタの設定

表 1 に本評価ボードでのメモリ・コンフィグレーション・レジスタの設定を示します。基本的にこれらの設定は、フラッシュ ROM 内にあるモニタ・プログラムの初期化部分で設定されるので、SDRAM にダウンロードして実行するユーザ・プログラムで初期化する必要はありません。しかし、完成したプログラムをフラッシュ ROM に書き込んで実行する場合は、これらのレジスタを初期化しないと、SDRAM や後述するポート機能が使えません。

MDCNFG は SDRAM のコンフィグレーションを行うレジスタです(図 2)。SDRAM バンク 0/1 側と 2/3 側の設定が可能です。ここでは、各 SDRAM バンクの有効/無効、バス幅、ロウ/カラム・アドレス幅、バ

表 1 メモリ・コンフィグレーション・レジスタの設定値

レジスタ名	アドレス	設定値
MDCNFG	0x48000000	0x1AA1
MDREFR	0x48000004	0x03C110C8 → 0x038190C8
MSC0	0x48000008	0x23F023F0
MSC1	0x4800000C	初期値 (未使用)
MSC2	0x48000010	初期値 (未使用)
MECR	0x48000014	初期値 (未使用)
SXCNFG	0x4800001C	初期値 (未使用)
SXMRS	0x48000024	初期値 (未使用)
MCMEM0	0x48000028	初期値 (未使用)
MCMEM1	0x4800002C	初期値 (未使用)
MCATTO	0x48000030	初期値 (未使用)
MCATTI	0x48000034	初期値 (未使用)
MCI00	0x48000038	初期値 (未使用)
MCI01	0x4800003C	初期値 (未使用)
MDMRS	0x48000040	0x1900