

組み込みシステムにおけるオンチップ/オフチップメモリ アーキテクチャを対象とした省電力ページングアルゴリズム

小林 さとみ¹ 中西 恒夫² 福田 晃³

本論文では、組み込みシステムを対象とした省電力ページングアルゴリズムについて述べる。オンチップ上のメモリ空間を利用して、ページング機構を利用することによる省電力目的の動的なメモリ割り当てを行い、チップ内部のメモリを活用する手法である。

本ページング機構では、タイマ割込みによる周期的に起動されたメモリ管理の中で、過去のページ情報テーブルに格納されたプログラムの抽出された挙動情報から電力消費の低いメモリ割り当ての予測を行い、挿入ページと犠牲ページの候補を決定する。こうすることでページングにおけるオーバーヘッドを減少し、省電力性の効果を上げる。

Keywords: 組み込みシステム, 省電力, メモリ管理, ページング, オンチップ・オフチップアーキテクチャ

Power Aware Paging Algorithm of On-chip/Off-chip Memory Architecture for Embedded Systems

Satomi Kobayashi¹, Tsuneo Nakanishi², and Akira Fukuda³

This paper proposes a method of electric power aware memory management algorithm for embedded systems. It means the one of methods of internal memory utilization and realization of low energy system by dynamic power aware memory assignment of paging mechanism.

Our memory management is executed by timer interruption. It selects insertion page and victim one from predicting next memory power consumption of previous paging information table, in order to reduce paging overhead.

Keywords: embedded system, low power, memory management, paging, on-chip/off-chip architecture

1 はじめに

組み込み機器では消費電力が大きな課題になっている。その中でも大きな電力消費の割合を占めているメモリについては、HW/SWの双方からさまざまな研究がなされている。組み込み機器を構成するデバイスの機能も高度化し、メモリに関しても今後大きなメモリ空間を構築することが可能となる。また、メモリ上にファイルシステムを構築する研究もなされている様に [1], 大容量メモリの構成では従来と異なるモデルの構築が可能となる。メモリ管理にもアドレス空間の管理ばかりで

1, 帝塚山大学, 情報教育研究センター
Tezukayama University, Education Center of Information
Processing
kobayashi.satomi@tezukayama-u.ac.jp
2, 奈良先端科学技術大学院大学, 情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology
tun@is.aist-nara.ac.jp
3, 九州大学大学院, システム情報科学研究院
Graduate School of Information Science and Electrical
Engineering, Kyushu University
fukuda@f.csce.kyushu-u.ac.jp

なく、今後は他機能の管理も必要となる。そこで本稿では、大容量のメモリ空間を OS レベルで電力消費を下げる手法について提案する。

これまでの研究では、オンチップメモリに参照の多いページ置き、オフチップメモリに参照の少ないページになるよう移動させるページングを行うことによって、消費電力の削減を行う手法と評価について述べた。この方式は、外部回路の消費電力や、メモリ参照の頻度によっても評価が異なる。低消費電力を実現するページングアルゴリズムについて述べる。

このとき考察すべき課題は以下のとおりである。

- ・プログラムの挙動の把握

ページングを行う場合に、プログラムのアドレス参照状況を全て把握することはできない。例えばプログラム実行中のメモリ参照をどのように行い、それらをどのように評価するかという、プログラムの挙動の把握という問題がある。

- ・オーバーヘッドとのトレードオフ

メモリ管理に省電力の付加機能を持たせることで、実行時間が長くなり電力消費が高くなる可能性がある。オーバーヘッドを電力消費で考え、プログラムを実行する際にトータルの電力が低いことが証明されることが必要である。

2 関連研究

OS レベルでの省電力化手法アルゴリズムに関する研究は以下の様なものがある。

Lu らは、[3] で ACPI(Advanced Configuration and Power Interface) を利用した、PC 環境での HDD の省電力指向の最適化アルゴリズムを提案している。電力状態を読み書き中、スピニング中、スピニング停止中の 3 状態で切替え、状態が変化するタイムアウト値を設定して動的に行っている。セッション時間の予測値と実際のセッション時間の長さを設けることで最適化を図っている。

Lebeck らは、[4] では、ページ単位で外部メモリを管理しキャッシュのヒット率を上げる研究で、ページ入れ替えに関しては単純なヒストグラムで優先順位をつけている。また、外部メモリの省電力化には、状態遷移にかかる電力消費から閾値をあらかじめ求めて、動的にオーバーヘッドを算出することで変更している。

Qiu らは [5] で、統計的情報からマルコフ決定

プロセスと、確率的数学的なモデリングを行うことで予測を行っている。省電力サービスの提供者と要求者についての優先度をこれらの新しいモデリングを行うことで解決している。

また、アプリケーションに特化した消費電力マネージャの実装について、Marchal らが [6] で行っている。彼らは MPEG 用のデバイスドライバの実装を行った。彼らは、配送層、同期層、圧縮層、構成層 4 つのレイヤーに分割することで省電力のためのデバイスを開発した。この論文では、静的に制御/データの流れ解析を行ってメモリ参照予測を行っている。

3 対象とするハードウェアアーキテクチャ

3.1 メモリアーキテクチャ

対象システムは仮想記憶をサポートし、仮想記憶方式はページングとする。

対象とするアーキテクチャは、図 1 に示すように、1 つのチップには、プロセッサおよびメモリ(オンチップメモリ)が搭載されている。メモリ容量を拡大する目的で、オンチップメモリの他に、別チップとして、さらにメモリ(オフチップメモリ)がある。すなわち、メモリは、オンチップメモリとオフチップメモリから構成されており、オンチップメモリとオフチップメモリとをあわせて、1 つの線形な物理アドレス空間を構成する。

メモリアクセス時間は、オンチップメモリは 1 命令に対して 1 クロックを要する。また、オフチップメモリに対しては、load 命令発行時に 30 クロック、store 命令発行時に 10 クロック前後である。また消費電力については、オンチップメモリは、通常状態で電流の流れがない SRAM を使用しているため電力が低く、オフチップメモリは定期的に取りフレッシュしている DRAM を使用し、バス回路と DMA の駆動があるためより電力を消費する。

また、チップ間でページをコピーまたは移動する(オンチップメモリからオフチップメモリまたは、その逆)際には、DMA を用いる。このコピーまたは移動は、後述するページング機構で用いることになる。

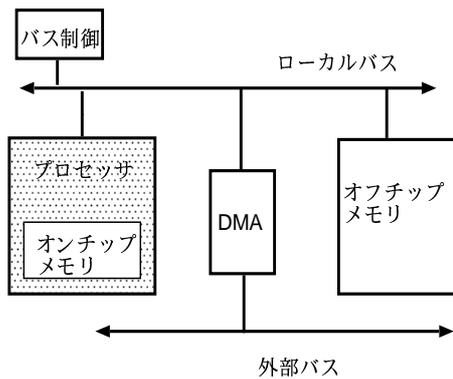


図 1: 対象とするアーキテクチャ。

3.2 電力モード

オフチップメモリの消費電力モードとして、いくつかのモードをハードウェアでサポートしているとする。本論文では、単純な場合である、high (通常モード) と low (省電力モード) の 2 つのモードとする。電力モードが high のときは、当該メモリは、アクセス可能であるが、消費電力が高いことになる。このモードは、従来から実現されている通常モードである。一方、low モードは、省電力のために設けられたモードである。オフチップメモリが low モードの場合、当該メモリの消費電力は少ないが、プロセッサからのメモリアクセスは行えず、OS に通知がいくものとする。オフチップメモリの電力モードを high にするか low にするかは、OS が設定できるものとする。

上記をプロセッサからのアクセスの観点からまとめると、以下となる。

- ・プロセッサがオンチップメモリへアクセスする場合は、そのままアクセスできる。
- ・プロセッサがオフチップメモリへアクセスする場合、当該メモリが high モードのときは、そのままアクセスできる。low モードのときは、アクセスができず割り込みみがかかると OS に制御が移る。

すなわちアプリケーションプログラムは high モード時にオフチップメモリ参照するとそのまま実行されるが、low モード時に同様のことを行くと、割り込みみがかかってプログラムは一旦停止する。そして OS の電力制御機構に制御が移る。そしてオフチップメモリの電力モードを high にした後、またアプリケーションに制御が移動し、メモ

リ参照が行われる。

4 提案方式

4.1 概要

本システムではマルチタスクを実現できるリアルタイム OS とする。すなわち OS は、複数のタスクを待ち時間の制約を満たしたリアルタイム性をもって実行可能なものとする。タスクの駆動要求がかかると、一定時間以内に必ず実行されるものとする。

そのためタスクは、実行可能状態、I/O 待ち状態、休止状態の 3 つの状態に遷移する。また、実行可能状態にあるときには、OS はタイムスライスでそれぞれのタスクに公平に CPU の割り当てを許可する。実行可能状態のタスクを、実行中とレディーの 2 つの状態をタイムスライスでスイッチしながら割り当てる。

また、メモリ管理についても一つのタスクとして実行され、タイマ割込みをうけ周期的に起動される。

論文 [2] では、ページ参照情報収集機構で、タスク実行時のメモリ参照状況をページ情報テーブルに格納している。これは、メモリ参照の状況を抽出した部分的な情報である。また電力制御機構では、一定周期に外部メモリ参照がなかった場合に省電力モードに切り替わり、省電力モードの時に外部メモリ参照が発生した場合に通常モードに切り替わるが、これらの情報もページ情報テーブルに格納される。

現実的に全ての参照状況をメモリに保存することは実現不可能である。タスク実行時の抽出された情報から省電力化に必要なページ予測が必要となる。

本稿では、メモリ管理が周期的に情報収集することを利用して、論文 [2] で行われた挿入ページの選択モジュールと、犠牲ページの選択モジュールにおけるページングアルゴリズムについて、タスク実行時のメモリ参照データ情報から消費電力の高いページをオンチップに入れ、低いページをオフチップにおくことで電力消費を低減させる。

ページ参照情報収集機構では、タスク実行時のメモリ参照のトレース情報を収集している。次の周期のメモリ参照を予測しページ入れ換えを行うかどうか評価する。

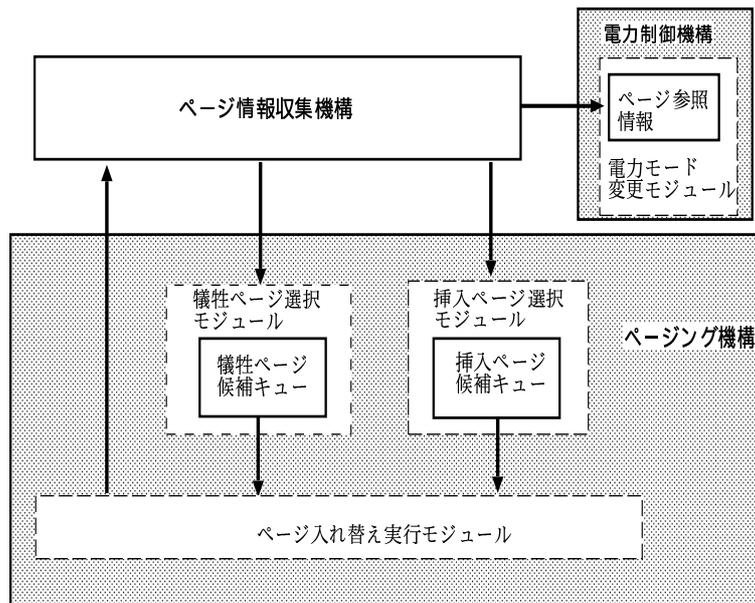


図 2: 省電力メモリ管理のモジュール構成.

キャッシュの場合にはメモリ参照されたデータが必ず内部に転送される。しかし、本ページングでは外部メモリ参照が疎な場合には、ページ移動を行わない。外部メモリ参照が密であるか疎であるかを、ページ情報収集機構からのデータで判断し、電力消費についての判定関数を用いて、移動時のオーバーヘッドを加えても省電力に効果がある場合には、ページ移動を行う。

4.2 メモリ管理の構成

省電力指向を備えたメモリ管理は、アプリケーション実行情報収集機能、ページ入れ替え候補キュー作成機能、ページ入れ替え判定機能、ページング実行機能に分かれる。

(1) アプリケーション実行情報収集機能: アプリケーションのメモリ参照状況についてのデータをページ情報収集機構のデータから収集する。すべてのメモリ参照を把握することは不可能なので、メモリ管理に必要な最小限の情報を格納する。そしてこれは、ページ入れ替えの優先度を定める材料となる。具体的には MMU と電力制御機構からの情報であり、ページ参照ビット、省電力モードビット、オンチップ/オフチップメモリビットである。

ページ参照ビットとは、プログラムがあるページを参照したときに必ず 1 になる領域である。また省電力モードビットとは、オフチップメモリが省電

力モードから通常モードに電力が変化するきっかけとなる参照ページに対する情報フラグである。またオンチップ/オフチップメモリビットとは、ページがオンチップ上にあるかオフチップ上にあるか識別するフラグである。

(2) 候補キュー作成機能: アプリケーション実行データから、ページ入れ替えの候補キューを作成する機能である。本稿では、過去のページ参照で「密なページ」と「疎なページ」を定義し、それにより順序づけを行う。

(3) ページ入れ換え判定機能: 選択された挿入ページと追い出しページでの候補を入れ換えることが電力のコストに見合うかどうか判定を行う。というのも、ページ入れ換えには大きなオーバーヘッドが伴うからである。この機能には、オーバーヘッドに関する消費電力プロファイルを持つ。

(4) ページング実行機能: (3) で判定された結果に基づいてページの入れ換えを行う。

4.3 候補キュー作成機能

4.3.1 省電力モードを利用した候補キューの作成

本機能には、メモリ参照解析機能がある (図 3)。オフチップメモリの参照が密の場合と疎の場合の定義について考える。

ページ情報収集機構のある周期でデータ参照

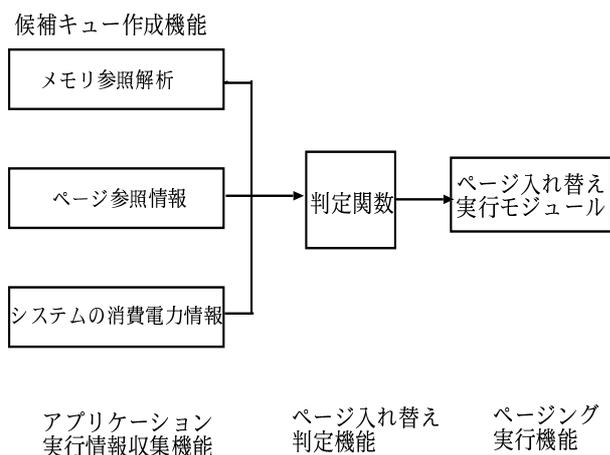


図 3: ページングの機能構成.

ビットが立っているページ群を $P[n]$ とし, $P[n]$ の中からデータ参照が疎であるページとデータ参照が密であるページを推測する手法を用いる.

参照履歴が残っているページについて, 外部メモリの省電力モードフラグから 3 つのステータスが考えられる.

- (1) 参照だけがされている
- (2) 参照はされたがその後省電力モードになった
- (3) 省電力モードである時に参照されて通常モードに変わる

上記のステータスについて, 過去の周期におけるページ参照の頻度については,

$$(1) \geq (3) \geq (2)$$

であるといえる.

なぜなら, 参照されて省電力モードになったページ, すなわち参照ビットと省電力ビットがオンになっているページは, 参照後に一定周期期間以上外部メモリ参照がなかったため省電力モードになったためである. また, 省電力モードであった時にページ参照が起こり通常モードに変わった場合は, 全く参照のない期間がしばらくあったが, 一定周期期間以上参照がなかったわけではな. (2) よりは密な参照を行っている可能性は高い.

このような方針で, 上の 3 つのステータスから外部メモリ参照に関する優先度を決め挿入ページ候補キューの中に格納する.

4.3.2 ページ参照情報からの候補キューの作成

本メモリ管理は周期的に起動され, n 回に 1 回の割合でページングが実行されるものとしている. したがって, ページングを実行しない場合は, ただ単にアプリケーション実行の情報収集のために呼び出される.

ページ参照情報からの候補キュー選定の手法について述べる. 最近データ参照があったページに対しては, 次の周期でもデータ参照が発生しやすいことを利用する. 現在の時刻からどれだけ離れた時刻にデータ参照があったかで優位性を決める. 現在の時刻に近い方のプライオリティを上げて評価する.

周期的に読み込んだ参照ビットから, どのページがどの周期の間で参照されているか, 当該ページを参照する頻度が高いかどうか判断する.

またこれまで参照ビットが何回立ったか. それぞれ現在の時刻からどれだけ離れた時間に行われたか, という 2 つのパラメータで評価し, 挿入すべきページを順序づける.

4.4 ページ入れ替え判定機能

全ての回路に利用時, 非利用時の電力モードがあることを前提としている. 組込みシステムを構成する回路の電力消費量は, プログラムの実行時間を $[t1, t2]$ とすると以下の式で表される.

$$f(t1, t2) = \int_{t1}^{t2} W(t)dt \quad (1)$$

この場合, 本システムでのトータルの電力消費量はプロセッサ, バス, DMA, オフチップメモリの総和で表され, 下記の様に表記できる. それぞれの回路には利用時, 非利用時の消費電力が存在し, (それぞれ W_{high}, W_{low}) この値を最小にすることで求められる.

$$TotalEnergy = E_{CPU} + E_{Mem} + E_{DMA} + E_{Bus} \quad (2)$$

$$E_{Mem} = W_{high} \times Cycles_{high} + W_{low} \times Cycles_{low}$$

また, システムの消費電力情報として, 以下のモジュールの実行における電力消費量をオーバーヘッドのパラメータとして持ち, 本機能で利用する.

- ・ ページング管理モジュール
- ・ ページ入れ替え実行
- ・ オフチップメモリの電力モードの切替え
- ・ オフチップメモリのランダム参照
- ・ DMA とバス制御回路の駆動

上記の省電力モードビットからの推測と、ページ参照ビットからの推測の双方から、ページ入れ換えの候補を決め、前の周期におけるページ参照状況とページングを行った場合のページの参照状況から電力消費を求め、入れ換えを下式を満たすかどうかで判定する。

$$\{ \text{ページ移動実行時電力消費} \} < \{ \text{ページ移動非実行時の電力消費} \} \quad (3)$$

例えば $P[k1], P[k2], P[k3]$ というページ参照が直前の周期であったとき、 $P[k3]$ が入れ換えの候補で、オフチップに出すページを $P[k4]$ とすると、ページ参照が $P[k1], P[k2], P[k4]$ となる場合の次の周期の電力消費を求め、後者の方が電力消費が低くなった場合には、「ページ入れ換え」と判定する。メモリ参照状況は、前周期での参照で見積もるものとする。ページングを実行する際に、ページ移動の候補キューにしたがって行うが、その移動の判定の際に評価式 (3) に従って決定する。

4.5 ページングアルゴリズム

これまでに述べたとおり、本省電力メモリ管理は、アプリケーションの挙動をページ参照状況という形で把握している。

ページ参照情報収集機構は一定周期 T ごとに起動され、アプリケーションのページ参照情報を入力する。ページ参照収集周期を n とするとページングの周期 T とは下記の関係がある。

$$T = n \times$$

そして、一定期間、一定の回数それらの情報を収集して、メモリ参照解析を行い、プライオリティを付けて挿入ページ候補キューと犠牲ページ候補キューに保存する。

また、ページ入れ替えの判定関数では、参照の頻度ではなく、消費電力に項目をしぼって、ページ入れ替えの際の判定を行っている。

さらにマルチタスクの場合には、タスクに対してページ番号を割り当てることで実現できる。ペー

ジ参照されたページ番号をタスクのレディーキューの順序に従って順序づけを行う。スケジューラのレディーキューから次の周期におけるタスクとページ参照を予測し、当該タスクについてのページ移動を優先する。

挿入ページの候補と犠牲ページの候補を前章に従って求める。各ページキューの先頭からページ番号を評価していくが、消費電力のコストが該当しない場合には、入れ替えを見合わせる。

追い出しページの候補キューの生成に関しては、挿入ページの選択モジュールの逆の評価を行うことによって実現する。すなわち、前項目の式 (3) で不等号を逆にして行う。

ページングアルゴリズムを要約すると以下のとおりになる。

- 1) タイマ割込みによりメモリ管理が起動される。
- 2) ページ参照テーブルを読み、 $P[n]$ に参照ページ番号を格納
- 3) n 周期分実行する
- 4) 省電力モードビットから $P[n]$ の優先順位を決める
- 5) ページ参照ビットから $P[n]$ の優先順位を決める
- 6) 挿入ページ候補キューを作成
- 7) 犠牲ページ候補キューを作成
- 8) 入れ替え候補を決める
- 9) ページ入れ替えの判定条件 (3) 式を満たした場合にページ入れ替えを実行。
- 10) (3) 式を満たさない場合には何も行わない
- 11) 全ての配列をクリアし終了

5 おわりに

オンチップ/オフチップアーキテクチャにおける省電力メモリ管理の機能についてアルゴリズムを述べ提案を行った。今後の課題は、これらの機能について検証し評価を行うことである。

6 謝辞

本研究の一部は科学技術振興事業団戦略的基礎研究推進事業 (CREST) の支援のもとに行われたものである。

参考文献

- [1] 石墨 紀孝, 最所 圭三, 福田 晃: ”組み込みシステム向けフラッシュメモリファイルシステムの設計”, 信学論, pp.90-99, 2001.
- [2] 小林さとみ, 中西恒夫, 福田晃: ”ハードウェアサポートの省電力モードとページング機構を利用した省電力メモリ管理”, コンピュータシステムシンポジウム論文集, pp.17-23, 2001 .
- [3] Y.H.Lu, T.Simunic, G.Micheli: ”Software Controlled Power Management”, *Proceedings of the 7th International Workshop on Hardware/Software Codesign*, pp.157-161,1999.
- [4] A.R.Lebeck, X.Fan, H.Zeng and C.Ellis: ”Power Aware Page Allocation”. *Proceedings of the Architectural Support for Programming Languages and Operating Systems*, 2000.
- [5] Q. Qiu, Q. Wu and M. Pedram, ”Stochastic Modeling of a Power-Managed System: Construction and Optimization”, *Proceedings of ISLPED*, pp.194-199, 1999.
- [6] P.Marchal, C.Wong, A.Prayati, N.Cossement, F.Catthoor, R.Lauwereins, D.Verkest, and H.De Man: ”Dynamic Memory Oriented Transformations in the MPEG4 IM1-Player on a Low Power Platform”, *PACS 2000*, pp.40-50, 2001.