

消費電力見積もりが可能な不揮発性 CPU シミュレータの試作

Prototype implementation of non-volatile CPU simulator with power consumption estimation facility

山田 晋平†
Shimpei Yamada

趙 茜†
Qian Zhao

中本 幸一†
Yukikazu Nakamoto

辻 幸秀‡
Yukihide Tsuji

根橋 竜介‡
Ryusuke Nebashi

1. はじめに

筆者らは、不揮発性素子を利用した CPU において、消費電力を制御する方式を提案している。本報告では、この方式を評価するための CPU シミュレータの試作を報告する。本 CPU シミュレータでは既提案の消費電力を制御する機構を実装し、揮発性素子、不揮発性素子、両者の消費電力、待機電力を見積もりが可能である。シミュレータの作成にはオープンソースソフトウェアである SkyEye^[1]を用い、MSP430 を対象としている。本不揮発性 CPU の応用領域として、センサノードを想定している。このため、センサ・アクチュエータをシミュレートするために、シミュレータはセンサデバイスシミュレータを付加する機能を有する。

2. シミュレータ概要

ノーマリオフの特性を考慮したセンサの動作をシミュレートするため、センサに用いるプロセッサとセンサデバイスのシミュレーション環境を構築する。この環境をセンサシミュレータとし、その構成を図 1 に示す。

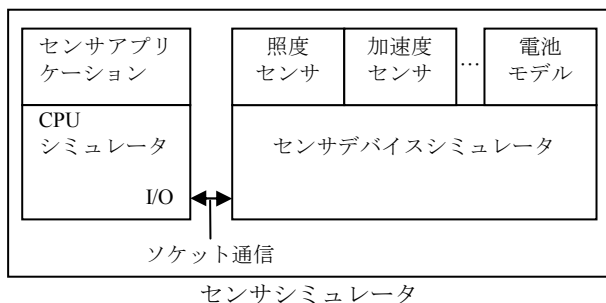


図 1 センサシミュレータの構成

CPU シミュレータは対象とするプロセッサをシミュレートし、センサとしての動作を行うセンサアプリケーションを実行する。センサアプリケーションは I/O への読書きを介してソケット通信を行い、センサデバイスシミュレータとセンサ情報をやり取りする。センサデバイスシミュレータは各種センサデバイスや電池モデルをシミュレートする。

3. CPU シミュレータ

CPU シミュレータは、汎用の命令セットシミュレータである SkyEye を用いて実装する。SkyEye は Linux や Windows 上で動作し、ARM, ColdFire, Mips, PowerPC,

† 兵庫県立大学大学院 応用情報科学研究科, UH

‡ NEC グリーンプラットフォーム研究所, NEC

SPARC, x86, Blackfin などのプロセッサをサポートしている。また、GDB サーバモードで起動することにより GDB を用いたデバッグが可能である。

SkyEye のソースは C 言語で記述されており、プロセッサ依存部を規定のインタフェースに基づいて実装することで、新たなプロセッサに対応可能である。プロセッサ依存部はアーキテクチャ実装部 (以下、arch と呼ぶ) とチップ仕様実装部 (以下、mach と呼ぶ) に分けて扱われる。今回は arch を MSP430 とし、その依存部を実装する。mach については特定のチップを想定せず、仮想的なチップ msptest を定義して扱うものとする。

SkyEye ではシミュレートするプロセッサやメモリ構成といった環境を指定するため、専用のコンフィグファイルが用いられる。コンフィグファイルには行毎に「オプション名: 引数」の形式で設定を記述する。引数が複数存在する場合はカンマで区切る。主な設定項目として、arch, mach, mem_bank (メモリ領域の種類, アドレス, サイズ) などがある。コンフィグファイルの記述例を図 2 に示す。

```
arch: msp430
mach: msptest
mem_bank: map=L, type=RW, addr=0x00000000, size=0x0200
mem_bank: map=M, type=RW, addr=0x00000200, size=0x7e00
mem_bank: map=F, type=RW, addr=0x0000c000, size=0x4000
```

図 2 コンフィグファイルの記述例

SkyEye ではシミュレートする CPU がメモリ空間にアクセスするとき、図 3 のようにアクセス先の種類に応じた関数が呼出し分けられ、読書きの処理が行われる。メモリ領域の種類とアドレスの対応はコンフィグファイルで指定された mem_bank の指定に従う。各領域の読書き時に呼び出される関数は、規定の構造体に登録することで SkyEye から呼出し可能となる。

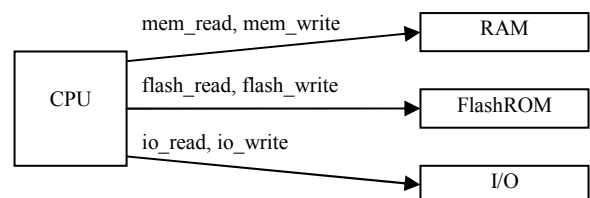


図 3 メモリ領域に応じた関数の呼出し分け

4. デバイスシミュレータ

4.1 追加する処理

プロセッサとセンサ間の通信をシミュレートするため、プロセッサの I/O 空間を介してソケット通信を行う仕組みを実装する。実装は mach に対して行い、以下の mach 用関数に処理を追加する。

- `msptest_mach_init`
SkyEye に対して `mach` を情報を登録する初期化関数。ソケット通信関係の初期化もこの関数で行う。
- `msptest_io_read_halfword`
コンフィグファイルで I/O 指定されたメモリ領域 (メモリマップド I/O) からの、2 バイト読み込み発生時に呼び出される。今回はメモリマップド I/O のアドレスとソケットを対応付けて通信を行うため、この関数では読み込みアクセスされたアドレスがどのソケットに対応付けられているかを調べ、対応付けられたソケットがデータを受信するのを待つ。
- `msptest_io_write_halfword`
`msptest_io_read_halfword` と同様に、この関数では書き込みアクセス先の I/O アドレスが対応付けられたソケットに対して、書き込みデータを送信する。

4.2 I/O とソケットの対応付け

メモリマップド I/O のアドレスとソケットの対応付けを SkyEye の外部から指定できるよう、コンフィグファイルにオプション `io_sock` を追加した。追加したオプションの引数は以下の通りである。

- `type=R` (読み込み) / `W` (書き込み) (アクセス方法指定)
- `addr=0xXXXXXXXX` (I/O アドレス指定)
- `ip=XXX.XXX.XXX.XXX` (IP アドレス指定)
- `port=XXXXX` (ポート番号指定)

MSP430 では、メモリ空間の `0x0100~0x01FF` 番地に 16 ビットペリフェラルモジュールが割り当てられており、コンフィグファイルにおける I/O のアドレスはこの範囲を指定する。4.2 節で説明した I/O アクセス時に実行される `msptest_io_read_halfword`, `msptest_io_write_halfword` はこの対応付け情報を参照し、アクセス先のアドレスに対応したソケットを用いて通信を行う。

4.3 センサシミュレータの実装例

センサ利用の例として、照度センサが部屋の照度を測定し、それを基地局がモニタリングする構成を考える。照度センサ内には、照度を実際に測定する照度センサデバイスと、そのセンサ値を処理するための CPU が載せられており、CPU 上で動作するセンサアプリケーションが、一定時間間隔で取得したセンサ値を基地局に送信するものとする。このような構成をシミュレートするため、図 4 のように照明と照度センサデバイス、CPU、基地局のシミュレータを作成した。

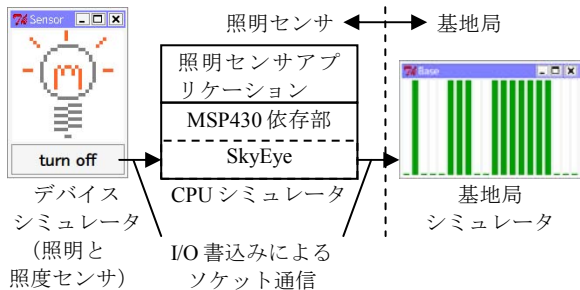


図 4 センサシミュレータ実装例の構成

デバイスシミュレータは照明と照度センサをシミュレートする。照明 (GUI のスイッチとランプ) の照度 (オン/オフの 2 値) を照度センサが測定しており、測定した情報を CPU シミュレータに送信する。CPU シミュレータ上

で動く照明センサアプリケーションは、受信したセンサ値を処理して基地局シミュレータに送信する。基地局シミュレータは受信したセンサ値を時系列でグラフ表示する。デバイスシミュレータと基地局シミュレータの実装は、開発、利用を容易にするため Ruby/Tk を用いて行った。

5. 消費電力の見積もり

本 CPU シミュレータは、MSP430 の命令実行における消費電力の見積もりに加えて、不揮発性 RAM^[2]利用時の消費電力を見積もる機能を有する。本 CPU シミュレータの構成を図 5 に示し、各部について説明する。

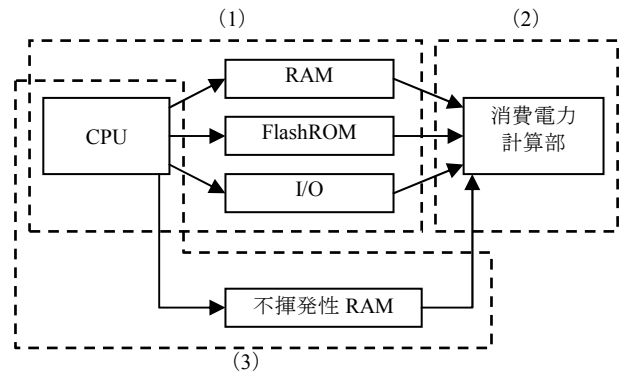


図 5 消費電力見積もり可能な命令シミュレータの構成

(1) SkyEye に MSP430 プロセッサ依存部と各メモリ領域へのアクセス用関数を実装することで MSP430 の命令をシミュレートする。

(2) 消費電力計算部を実装することで、各メモリ領域へのアクセスに応じた消費電力を計算する。

(3) 不揮発性 RAM を実装し、その読書きを制御するための不揮発性素子書き込みモードビットと不揮発性素子制御命令^[2]を、MSP430 プロセッサ依存部に追加することで、不揮発性 RAM 利用時の消費電力を計算する。

6. まとめ

本報告では SkyEye を使い、揮発性素子、不揮発性素子、両者の消費電力、待機電力の見積もりが可能な CPU シミュレータを実装した。今後はこのシミュレータを用いて実際のアプリケーションの動作をシミュレートし、消費電力削減手法の検討を行う予定である。

謝辞

この成果の一部は、独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) との共同研究業務であるノーマリオフコンピューティングプロジェクトの結果として得られたものである。

参考文献

[1] SkyEye, <http://www.skyeye.org/>.

[2] 根橋竜介, 辻幸秀, 崎村昇, 渡邊義和, 壬生亮太, 森岡あゆ香, 宮村信, 杉林直彦, 中本幸一, "不揮発性 CPU を用いた待機電力ゼロの電子システムの検討," 情報処理学会 組込システム研究会 (ETNET), 情処研報 2012-EMB-24, No.25, Mar 2012.