

平成21年度戦略的基盤技術高度化支援事業

「伝送モデルを統合するシステム・レベル電力・性能シミュレーションに  
よるアーキテクチャ探索システムの研究開発」

研究開発成果等報告書

平成22年 3月

委託者 関東経済産業局  
委託先 ケイレックス・テクノロジー株式会社



この印刷物は、印刷用の紙へ  
リサイクルできます。

## 目 次

|                                             |    |
|---------------------------------------------|----|
| 第1章 研究開発の概要.....                            | 4  |
| 1－1 研究開発の背景・研究目的及び目標.....                   | 4  |
| 1－2 研究体制.....                               | 5  |
| 1－3 成果概要.....                               | 7  |
| 1－4 当該研究開発の連絡窓口.....                        | 7  |
| 第2章 本論.....                                 | 8  |
| 2－1 伝送路モデルの開発.....                          | 8  |
| 2－2 システムレベルシミュレータへの統合.....                  | 12 |
| 2－3 アプリケーションソフト動作時のシステム電力評価.....            | 20 |
| 2－4 アーキテクチャ探索システムの実装組合せ制約条件に基づく網羅方法の検討..... | 24 |
| 2－5 アーキテクチャ探索システムの開発.....                   | 27 |
| 2－6 アーキテクチャ探索システムのテスト・検証.....               | 32 |
| 2－7 実測TEGボードの作成・測定.....                     | 36 |
| 第3章 全体総括.....                               | 44 |
| 3－1 研究開発成果.....                             | 44 |
| 3－2 課題.....                                 | 45 |
| 3－3 事業化展開.....                              | 45 |

## 第1章 研究開発の概要

### 1-1 研究開発の背景・研究目的及び目標

現在の携帯情報端末は、多機能、高機能、大容量、高速化、小型化、低消費電力化が求められている。

また製品サイクルも短いため、川下製造事業者等にとっては高品質のものを、短期間に開発する必要がある。高品質のものを短期間に開発するためのポイントの1つとして、フロントローディングにより後戻り工程を少なくすることが重要になっている。つまりフロントローディングにより後工程に影響する事柄を開発初期段階で検証しておくことで、後戻りを少なくすることが出来る。そのための技術としてシミュレーションがある。これは本来、製品ができた段階で実機にて検証することを、前もってコンピュータ上のソフトウェアにて動作させて、機能や性能を検証する手段である。

携帯情報端末等では、試作品を作成してからの機能や性能の検証で問題が発生すると、多大な後戻りとそれにとまらうコストや機会損失が発生する。従来技術として、製品開発の各工程でシミュレーションを使った検証は行われているが、システムレベルでのシミュレーションは機能に限定されている。システムレベルの消費電力検証については、各部品のカタログベースの電力性能の単なる積み重ねで行われている。そのためアプリケーション動作に対応する正確な消費電力は見積もれず、消費電力の評価は試作品の完成段階まで分からない。このような手法では製品設計が完成した段階での消費電力検証であるため、消費電力上の問題が発生した場合の後戻りは多大な損失に繋がる。

しかしながら、今日では製品の低消費電力化の重要性は益々高まっており、製品の電力性能はその製品価値として無視できなくなっている。即ち、システムレベルでの消費電力シミュレーションを製品設計の初期段階に行うことの重要性が非常に高まっている。

LSI チップ単体については、従来、その詳細設計が完了した段階で論理ゲートあるいはトランジスタ・レベルの回路を用いて行われている。しかし、最終製品のシステムとしてハードウェアを構成し、またその上でアプリケーション・ソフトウェアを動作させて消費電力のシミュレーションを行う環境はなかった。

更に、電子部品の高密度な実装方法として SiP 等も多くなってきたおり、システムレベルの消費電力に対する電子部品間の配線接続の影響も無視できなくなっている。電子部品間のインターフェイスに消費される電力は、LSI チップの全消費電力の 10-30%と小さくなく、またチップ実装方法により配線容量は数十倍以上になるため、実装方法を考慮したチップ間の配線を伝送モデルとして、システムレベルでの消費電力を正確にシミュレーションする必要がある。即ち、電子部品の構成方法として、(1)IP を SoC 内で集積、(2)LSI チップとしてプリント配線基板上で集積、(3)SiP としてワイヤボンディングによる集積、(4)3 次元積層 LSI として貫通電極(TSV)で集積、など実装方法や実装上の配置に依存する電子部品間の配線を考慮してシミュレーションする環境が必要である。

本開発では従来は個別に開発手法が検討されてきた LSI 設計、プリント基板設計及び半導体の高密度実装に関わる配線の伝送路モデル、ソフトウェア開発などを横断的に取り扱うシステムレベルの消費電力シミュレーション環境の開発と、さらにこのシミュレーション環境を用いて最適なシステム構成を探索可能とすることを目標とする。

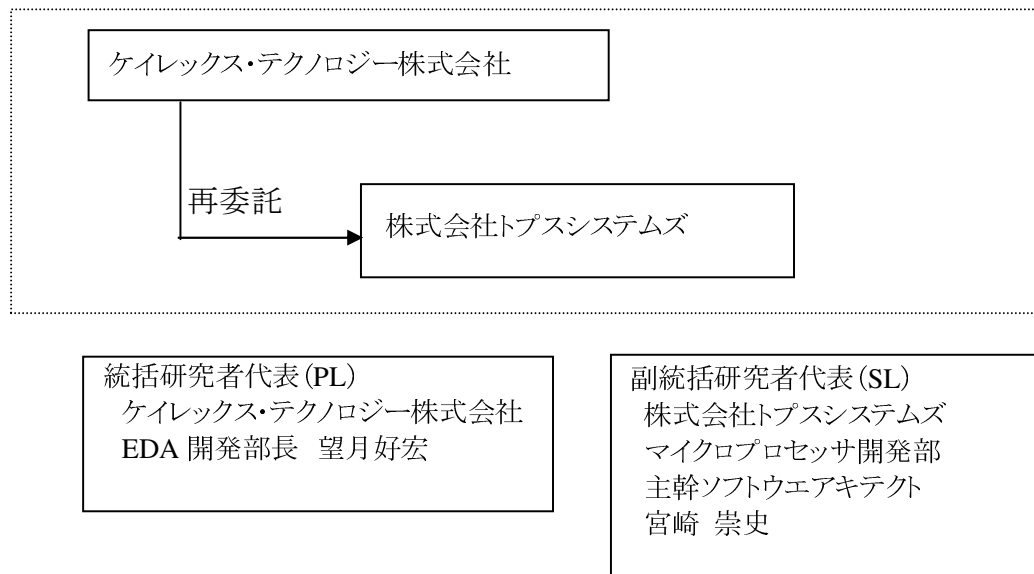
本研究の具体的な目標は以下のとおりである。

- (1) システムレベルの消費電力シミュレーション性能として、以下を目標とする。
  - ー消費電力シミュレーション精度:  $\pm 10\%$ 以内
  - ーシミュレーション速度: 1M命令/秒以上
- (2) 最適なシステム構成の探索を可能とする。

## 1-2 研究体制

以下に研究組織及び管理体制を示す。

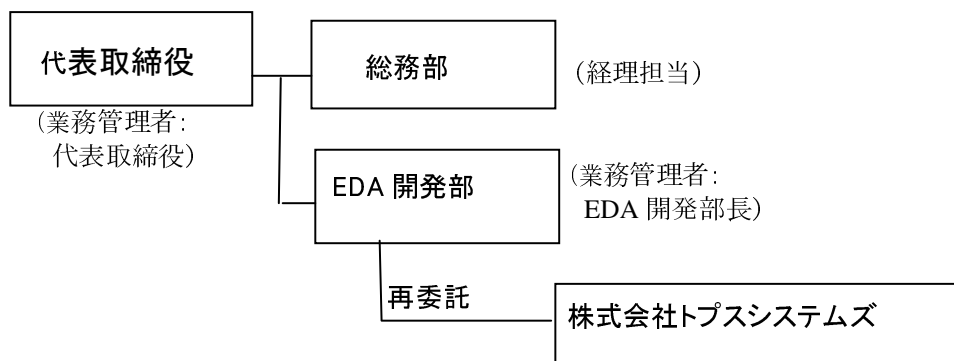
### 1-2-1 研究組織(全体)



### 1-2-2 管理体制

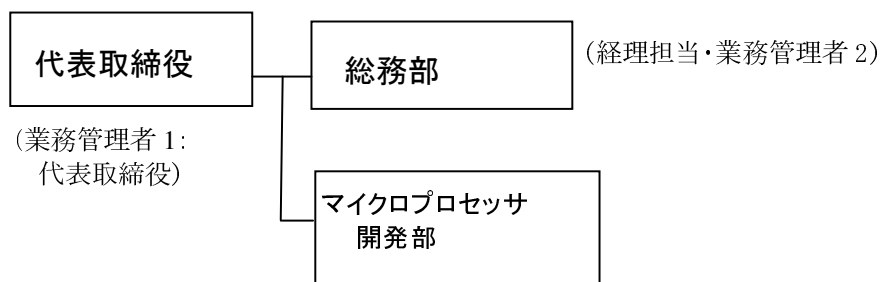
#### (1) 事業管理者

[ケイレックス・テクノロジー株式会社]



#### (2) 再委託先

[株式会社トプスシステムズ]



### 1-2-3 管理員及び研究員

#### 【事業管理者】ケイレックス・テクノロジー株式会社

##### □ 管理員

| 氏名    | 所属・役職              |
|-------|--------------------|
| 小篠 隆宏 | 代表取締役              |
| 望月 好宏 | EDA 開発部長           |
| 浅田 正紀 | EDA 開発部プロジェクトマネージャ |

##### □ 研究員

| 氏名       | 所属・役職              |
|----------|--------------------|
| 望月 好宏(再) | EDA 開発部長           |
| 板橋 幹人    | EDA 開発部プロジェクトマネージャ |
| 浅田 正紀(再) | EDA 開発部プロジェクトマネージャ |
| 岡田 典英    | EDA 開発部            |
| 河野 甲慈朗   | EDA 開発部            |

#### 【再委託先】

##### □ 研究員

株式会社トプスシステムズ

| 氏名    | 所属・役職                                |
|-------|--------------------------------------|
| 松本 祐教 | 代表取締役                                |
| 宮崎 崇史 | マイクロプロセッサ開発部<br>主幹ソフトウェアキテクト         |
| 大川 猛  | マイクロプロセッサ開発部<br>主幹 SW プラットフォームアーキテクト |
| 萩本 有哉 | マイクロプロセッサ開発部<br>ハードウェア/ソフトウェア技術スタッフ  |
| 磯部 正幸 | マイクロプロセッサ開発部<br>主任ソフトウェアキテクト         |

#### 【アドバイザー】

富士通マイクロエレクトロニクス株式会社

基盤商品事業本部 自動車事業部事業部長

独古 康昭 様

### 1-3 成果概要

今回の開発目的は、以下の2点であった。(3年計画のうちの初年度)

- ・ シミュレーション速度:1M命令/秒以上のシミュレーション環境構築
- ・ 最適なシステム構成を探索可能な、システム環境構築

上記の目的をクリアするために、まず実機測定用の実測 TEG ボード作成を行った。作成した実測 TEG ボードは伝送路のタイプの異なる以下のものを作成し、実測値による違いを確認した。

また市販の車載マイコンボードや FPGA ボードを用いて、伝送路の消費電力も測定した。

- － 実測 TEG ボード-1  
POP メモリ使用の測定ボード
- － 実測 TEG ボード-2  
FLASH/LPDRAM メモリ使用のバス幅が 16bit の測定ボード
- － 実測 TEG ボード-3  
FLASH/LPDRAM メモリ使用のバス幅が 32bit の測定ボード

これらの実測 TEG ボードを使って、実際に伝送路上での消費電力を測定した。また測定に際しては3つのアプリケーションにてそれぞれの消費電力を測定することが出来た。またこれらの結果から伝送路の違い、実行するアプリケーションの違いによる消費電力に違いがあることも確認できた。

さらに、異なる部品で構成された複数のシステムのシミュレーションを自動で実行し、最適なシステム構成のシステムを探索するソフトウェアの作成を行った。

本ソフトウェアでは、VisualSim を消費電力シミュレータとし、部品についてのシミュレーション情報、モデル情報、制約情報、その他検索条件として使いたい情報を指定することで、最適なシミュレーションを実行するような環境を作成し、単純な総当たりによるシミュレーション実行ではない効率的な実行ができた。シミュレーション実行速度目標(シミュレーション速度:1M 命令/秒)は 7M 命令/秒以上の結果を得ることができ、目標値をクリアできた。

また多数のシミュレーション結果を使って、最適なシステム構成のシステム検索ができることも確認できた。

特に制約情報の付加によるシミュレーション数の削減は、現実的なシステムにおいて非常に有効な設定となった。またその他の情報として、今回は部品のコストを設定できるようにしたことで、よりシステム構成の検討において有効なシステムとなった。

部品の性能、コスト以外でシステム要件の指標になりうる項目はあり、これについても現実的な検索条件付加の機能を備えたことで、システムとしての拡張性を持たせることができ、非常に有益なアーキテクチャ探索システムを開発できた。

(※VisualSim は米国 MirabilisDesign 社の登録商標である)

### 1-4 当該研究開発の連絡窓口

住 所: 東京都中央区入船 1 丁目 1 番 26 号永井ビル  
名 称: ケイレックス・テクノロジー株式会社  
氏 名: 望月 好宏  
所属役職: EDA 開発部長  
Tel: 03-3537-1925  
Fax: 03-3551-7622  
E-mail: mochizuki@keirex.com

## 第2章 本論

### 2-1 伝送路モデルの開発

#### 2-1-1 目的

本伝送路モデルの開発は、携帯情報端末等のシステム開発者が、開発するシステム全体に関する全ての電子部品(LSI チップ等のデバイス)と電子部品間の伝送路を考慮したシミュレーションにより、定量的な指標に基づいて性能と電力のトレード・オフを行うことで、全体最適化を可能とする開発環境を提供することを目的とする。

#### 2-1-2 目標

伝送路を含むシステム全体の消費電力シミュレーション環境として、高い電力シミュレーション精度と高速のシミュレーションとを目標とする。

＜システム全体でのシミュレーション性能＞

ー消費電力シミュレーション精度:  $\pm 10\%$  以内(実測値を基準とする)

ーシミュレーション速度: 1M 命令/秒以上 (システム内の CPU の命令実行速度)

上述のシステム全体のシミュレーションに対する目標を達成するために、伝送路のモデルとしての目標を設定する。

＜伝送路のシミュレーション性能＞

ー電力シミュレーション精度:  $\pm 10\%$  以内(実測値を基準とする)

ーシミュレーション速度: 1M トランザクション/秒以上

#### 2-1-3 伝送路モデル化手法

一般に、シミュレーション精度とシミュレーション速度とはトレード・オフの関係にあり、シミュレーション精度を向上させるとシミュレーション速度は低下する。シミュレーション精度を左右するのは、シミュレーション・モデルの抽象度である。

システムの動作をシミュレーションする際に、システムを構成する電子部品(デバイス)とそれらを接続する伝送路の2つに大きく分類することで、システム全体のシミュレーション・モデルを電子部品モデルと伝送路モデルという概念に分けることができる。そして、電子部品モデルと伝送路モデルのそれぞれの抽象度を定義することで、シミュレーション精度とシミュレーション速度とのトレード・オフを行う(図 2-1-1)。図の右上にいくほど、シミュレーション精度が向上し実測値に近づくが、シミュレーション速度は低下する。逆に、図の左下にいくほど、シミュレーション速度は向上するが、シミュレーション精度は低下する。

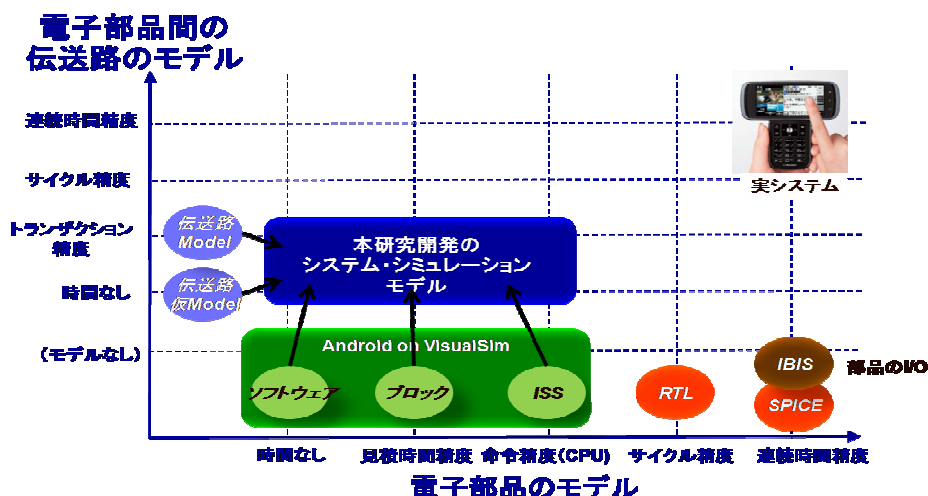


図2-1-1 シミュレーション・モデルの抽象度

本研究開発では、アプリケーション・ソフトウェアを動作可能な高速シミュレーションと±10%以内の精度を両立させるための基礎となる伝送路モデルを作成する。

伝送路モデルは、モデル化する対象により2種類のモデル化を行う。(表 2-1-1 参照)

表 2-1-1 本研究開発で作成する伝送路モデルの種類

| 伝送路モデルの種類 | モデル化する対象       | 例                         |
|-----------|----------------|---------------------------|
| 伝送路モデル    | 電子部品(チップ)間の伝送路 | CPUとメモリ(SDRAM, Flash 等)の間 |
| 伝送路仮モデル   | 電子ユニット間*の伝送路   | LIN, CAN, Flex Ray 等      |

\*電子ユニット:ECU 等の、電子部品をプリント配線基板上に搭載した電子基板

尚、伝送路モデルは平成 22 年度以降の研究開発の中で精度向上を行う予定であり、今年度は測定結果収集が中心である。また伝送路仮モデルは次年度以降の方針を決めるための実験的モデルである。

伝送路は、以下の手法(表 2-1-2)に基づいてモデル化する。

表 2-1-2 本研究開発で作成する伝送路モデル化の手法

|               | 従来のモデル化手法                              | 本研究開発のモデル化手法                                                                                                                                                                      |                                                                             |
|---------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|               |                                        | 伝送路モデル                                                                                                                                                                            | 伝送路仮モデル                                                                     |
| 伝送路モデルの対象     | I/O ドライバ+伝送路                           | I/O ドライバ+伝送路                                                                                                                                                                      | I/O ドライバ+伝送路                                                                |
| モデルの抽象度       | トランジスタ回路レベル                            | トランザクション・レベル                                                                                                                                                                      | トランスファー・レベル                                                                 |
| ・消費エネルギーの表現方法 | $E = \sum (\alpha CV^2 f + VI_{leak})$ | $E = (P_{read} \times T_{read}) + (P_{write} \times T_{write}) + (P_{idle} \times T_{idle})$                                                                                      | $E = E_{perdata} \times D_{trans}$                                          |
| ・モデルの単位       | ・信号線単位                                 | ・接続バス単位                                                                                                                                                                           | ・接続リンク単位                                                                    |
| 備考            | SPICE、IBIS+ネットでのシミュレーション               | $P_{read}$ : Read 時の消費電力[mW]<br>$P_{write}$ : Write 時の消費電力[mW]<br>$P_{idle}$ : Idle 時の消費電力[mW]<br>$T_{read}$ : Read 時間[s]<br>$T_{write}$ : Write 時間[s]<br>$T_{idle}$ : Idle 時間[s] | $E_{perdata}$ : 転送に必要なエネルギー[mWs/bit] ※実測値を使用する<br>$D_{trans}$ : 転送データ量[bit] |

伝送路モデルの各基本ステートの消費電力の計算方法を表 2-1-3 に示す。

表 2-1-3 本研究開発で作成する伝送路モデルの消費電力計算方法

| ステート | 電力値[mW]    | 計算方法                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| リード  | $P_{read}$ | $P_{read} = \alpha_{bitswitch} \times P_{always\_read} + P_{leak}$<br>$\alpha_{bitswitch}$ : ビット反転率(予測値、又はシミュレーション値)[%]<br>$P_{always\_read}$ : リード時電力( $C_{wire} \times V_{i/o}^2 \times f_{bus}$ ) $\times n \times k_{read}$ : 静的に設定<br>$P_{leak}$ : リーク電力( $V_{i/o} \times I_{leak}$ ) $\times n$ : 静的に設定<br>$C_{wire}$ : 伝送路の配線容量[pF]<br>$V_{i/o}$ : I/O ドライバの電源電圧[V]<br>$f_{bus}$ : バスの動作周波数[MHz]<br>$I_{leak}$ : I/O ドライバのリーク電流[μA]<br>$n$ : バスの信号数[本]<br>$k_{read}$ : リード・サイクルの係数 |

|      |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ライト  | $P_{write}$ | $P_{write} \doteq \alpha_{bitswitch} \times P_{always\_write} + P_{leak}$ $\alpha_{bitswitch}$ :ビット反転率(予測値、又はシミュレーション値)[%]<br>$P_{always\_write}$ :ライト時電力( $C_{wire} \times V_{i/o}^2 \times f_{bus}$ ) $\times n \times k_{write}$ :静的に設定<br>$P_{leak}$ :リーク電力( $V_{i/o} \times I_{leak}$ ) $\times n$ :静的に設定<br>$C_{wire}$ :伝送路の配線容量[pF]<br>$V_{i/o}$ :I/Oドライバの電源電圧[V]<br>$f_{bus}$ :バスの動作周波数[MHz]<br>$I_{leak}$ :I/Oドライバのリーク電流[mA]<br>$n$ :バスの信号数[本]<br>$k_{write}$ :ライト・サイクルの係数 |
| アイドル | $P_{idle}$  | $P_{idle} = V_{i/o} \times I_{leak} \times n$ $V_{i/o}$ :I/Oドライバの電源電圧[V]<br>$I_{leak}$ :I/Oドライバのリーク電流[mA]<br>$n$ :バスの信号数[本]                                                                                                                                                                                                                                                                                                                                                         |

※伝送路のステートは、アクセスの種類に応じて拡張する。例えば、バースト・リード、バースト・ライト等。

伝送路モデルの各ステートの時間を計算する方法を表 2-1-4 に示す。

表 2-1-4 本研究開発で作成する伝送路の各ステート時間の計算方法

| ステート | 時間[s]       | 計算方法                                                                                                                                                                          |
|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| リード  | $T_{read}$  | $T_{read} = Cycle_{read} \times N_{read} / f_{bus}$ $Cycle_{read}$ :1 回のリードに要するバス・サイクル数[cycles]<br>$N_{read}$ :リード・トランザクションの回数[回](シミュレーション値)<br>$f_{bus}$ :バスの動作周波数[MHz]      |
| ライト  | $T_{write}$ | $T_{write} = Cycle_{write} \times N_{write} / f_{bus}$ $Cycle_{write}$ :1 回のライトに要するバス・サイクル数[cycles]<br>$N_{write}$ :ライト・トランザクションの回数[回](シミュレーション値)<br>$f_{bus}$ :バスの動作周波数[MHz] |
| アイドル | $T_{idle}$  | $T_{idle} = T_{sim} - T_{read} + T_{write}$ $T_{sim}$ :シミュレーション時間[s]                                                                                                          |

※伝送路のステートは、アクセスの種類に応じて拡張する。例えば、バースト・リード、バースト・ライト等。

各種配線の基本伝送路モデルには、次の伝送路パラメータ（表 2-1-5）を設定する。

表 2-1-5 各種配線の基本伝送路モデルのパラメータ

| パラメータ名         | 記号                   | 単位  | 設定方法 |       |                                                         |
|----------------|----------------------|-----|------|-------|---------------------------------------------------------|
|                |                      |     | 事前   | Sim 中 | パラメータのソース                                               |
| ビット反転率         | $\alpha_{bitswitch}$ | %   | ○    |       | 50%(確率に基づく)                                             |
|                |                      |     |      | ○     | 命令レベル・シミュレータから各信号の反転率を統計情報の1つとして取得                      |
| 伝送路の配線容量       | $C_{wire}$           | pF  | ○    |       | 1) プリント基板上の配線仕様<br>2) プリント基板上の配線の測定値                    |
| バスの動作周波数       | $f_{bus}$            | MHz | ○    |       | バスの仕様                                                   |
| バスの信号数         | $N$                  | 本   | ○    |       | バスの仕様                                                   |
| I/O ドライバの電源電圧  | $V_{i/o}$            | V   | ○    |       | 1) デバイスの I/O 電源電圧仕様(Vdd)<br>2) デバイスの I/O 電源電圧測定値(Vdd)   |
| I/O ドライバのリーク電流 | $I_{leak}$           | mA  | ○    |       | 1) デバイスの I/O ドライバ・リーク電流仕様<br>2) デバイスの I/O ドライバ・リーク電流測定値 |

|             |                    |  |   |  |                               |
|-------------|--------------------|--|---|--|-------------------------------|
| リード・サイクルの係数 | $k_{\text{read}}$  |  | ○ |  | 1) 動作タイミング仕様に基づく計算値<br>2) 測定値 |
| ライト・サイクルの係数 | $K_{\text{write}}$ |  | ○ |  | 1) 動作タイミング仕様に基づく計算値<br>2) 測定値 |

※ Sim 中:シミュレーション実行中の動作に応じて異なる値が設定される。

各種配線とは、プリント基板、フレキシブル基板、ワイヤボンディング、 bumps 接続、LSI チップ電極等である。

## 2-2 システムレベルシミュレータへの統合

本研究開発は、各デバイス(チップ)に加えて、各デバイス間の伝送路の消費電力をモデル化して、アプリケーション・ソフトウェア動作時のシステムのハードウェアの動きに応じた消費電力を、シミュレーションにより算出することで伝送路を含む消費電力と性能のトレードオフを可能にすることを目的とする。

ターゲット・システムと同様に、アプリケーション・ソフトウェアのみならずプラットフォーム(OS)を含むソフトウェアを動作させるために命令セットシミュレータを用いる。また、ハードウェアの動作を正確にシミュレーションするために、システムを構成する各電子部品(チップ)と各電子部品間の伝送路モデルを含むハードウェアの動作を表現するハードウェア・モデルをシステムレベル・シミュレータ上に構築した。

そして、これら2つのシミュレータを統合して、実アプリケーション動作時のターゲットシステムの消費電力シミュレーションを行った。

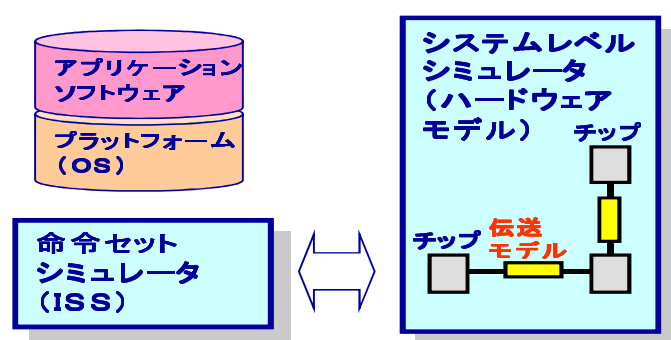


図2-2-1 命令セットシミュレータとハードウェアモデルの統合

### (1) 命令セットシミュレータの概要

命令セットシミュレータ(ISS: Instruction Set Simulator \*用語解説参照)は、マイクロプロセッサ(CPU:例えば、ARM、x86 など)の命令毎の動作を模倣するシミュレータであり、アプリケーション・ソフトウェアの機械語(\*用語解説参照)の命令列に基づいてマイクロプロセッサの動作を模倣する。

命令セットシミュレータを用いて、アプリケーション・ソフトウェア実行時にマイクロプロセッサが各電子部品にアクセスする状況を、アクセス統計情報として出力する。

命令セットシミュレータとして、オープンソースの QEmu(\*用語解説参照)を用いた。表 2-2-1 に、命令セットシミュレータの役割を示す。

表2-2-1 本研究開発における命令セットシミュレータの役割

| 項目       | 説明                                                                                                                                                                                                                                                                                                          |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 目的       | アプリケーション・ソフトウェアの命令レベルの動作に応じて、マイクロプロセッサから各電子部品へのアクセス情報を出力する                                                                                                                                                                                                                                                  |
| 使用シミュレータ | 命令セットシミュレータ QEmu(0.10.6)                                                                                                                                                                                                                                                                                    |
| 入力       | アプリケーション・ソフトウェア<br>(アプリケーション+OS のソフトウェアの機械語命令列)                                                                                                                                                                                                                                                             |
| 出力       | CPU から電子部品へのアクセス統計情報<br><br><ul style="list-style-type: none"> <li>・実行した命令数 (Instruction Count)</li> <li>・DDR_Read/Write 回数</li> <li>・Flash_Read/Write 回数</li> <li>・SD_Read/Write 回数</li> <li>・FB_Write 回数 (フレームバッファ:LCD 表示用)</li> <li>・Wifi_Send/Recv 回数(無線 LAN)</li> </ul> 尚、アクセス統計情報の出力は、10 万～100 万命令ごと。 |

命令セットシミュレータから出力した各電子部品へのアクセス統計情報を、各電子部品の性能・消費電力モデルに与えることで、システム全体の性能及び消費電力値を算出した。

## (2) システムレベルシミュレータの概要

性能及び電力のシミュレータとして、米国 MirabilisDesign 社の VisualSim を用いた。

ターゲット・システムの性能及び電力モデルは、ソフトウェアにより駆動されるハードウェアの動作を表現するビヘイビア・モデルと、各電子部品から構成されるアーキテクチャ・モデルから成る。

アーキテクチャ・モデルは、システムを構成する各電子部品と、それらの電子部品間を接続する伝送路からなる。各電子部品の性能と消費電力は、それぞれ性能パラメータと消費電力パラメータを持つ。

性能パラメータ：

- ・処理時間(nsec): 各電子部品が入力を受け取ってから処理結果を出力するまでの時間  
※伝送路については、処理時間(遅延時間)は規定しない。即ち、伝送路モデルがなくてもシステムの処理時間は適切にシミュレーションすることができる。

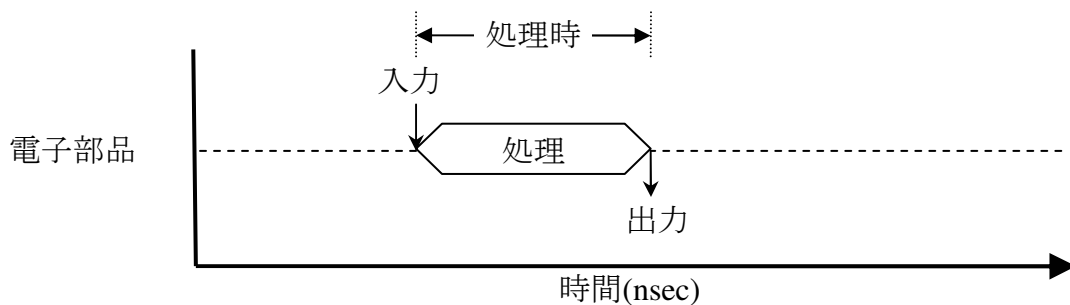


図2-2-2 各電子部品の性能パラメータ

消費電力パラメータ：

- ・Standby 状態の消費電力(mW): 各電子部品が Standby 状態にあるときの消費電力値  
※Standby 状態とは、当該電子部品が動作していない Idle 状態にあること。
  - ・Active 状態の消費電力(mW) : 各電子部品が Active 状態にあるときの消費電力値  
※Active 状態とは、当該電子部品が動作している状態にあること。
- 電子部品によっては、複数の動作があり、それぞれに消費電力値が異なるため、複数の Active 状態を規定する。

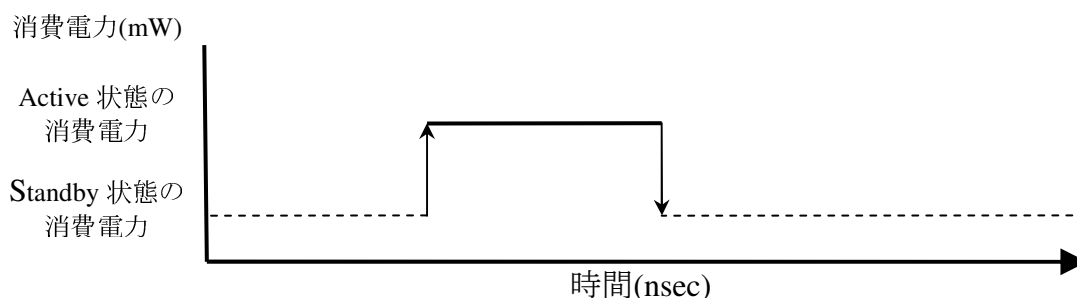


図2-2-3 各電子部品や伝送路の消費電力パラメータ

一方、ビヘイビア・モデルは、ターゲット・システム上の CPU 等のトランザクションのマスタとなる電子部品からアクセス先の電子部品へのアクセスに関連して、動作する各電子部品と各伝送路の動作とその経路を規定する。

システム全体の消費電力は、アーキテクチャ・モデルの各電子部品に規定した消費電力状態(消費電力値をパラメータとして設定)とその状態にある時間とから積算した。

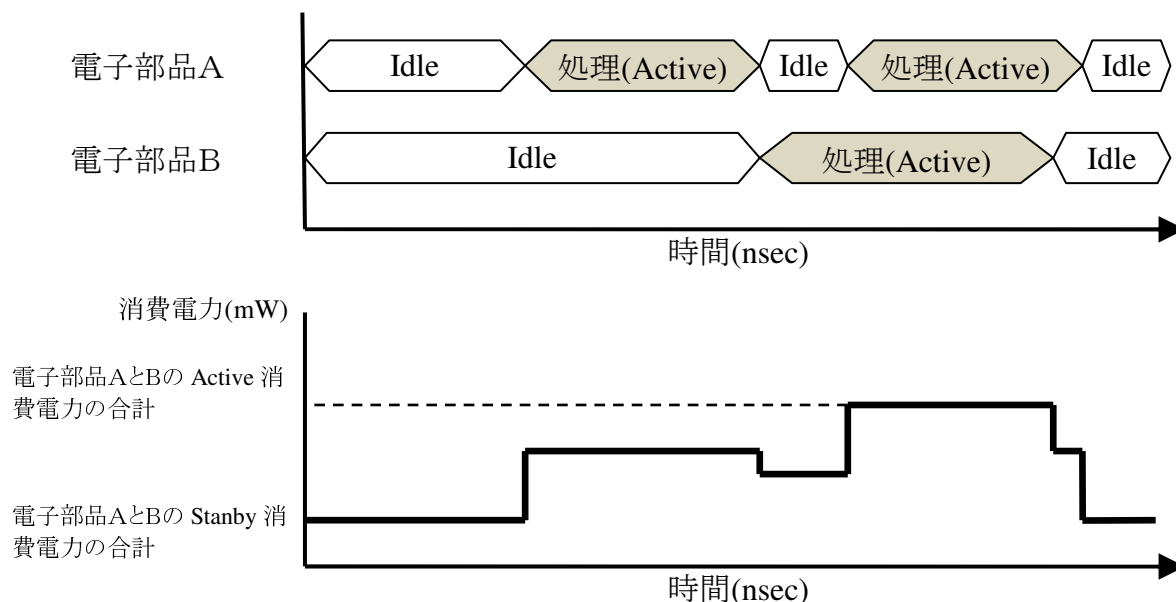


図2-2-4 各電子部品や伝送路の消費電力パラメータ

本研究開発では、OMAP(CPU と DSP を含むシステム LSI)を用いたシステムを対象に伝送路を含む性能と消費電力のシミュレーションを行った。図 2-2-5 に、本研究開発のターゲット・システムの性能・消費電力モデルの構成(画面写真)を示す。

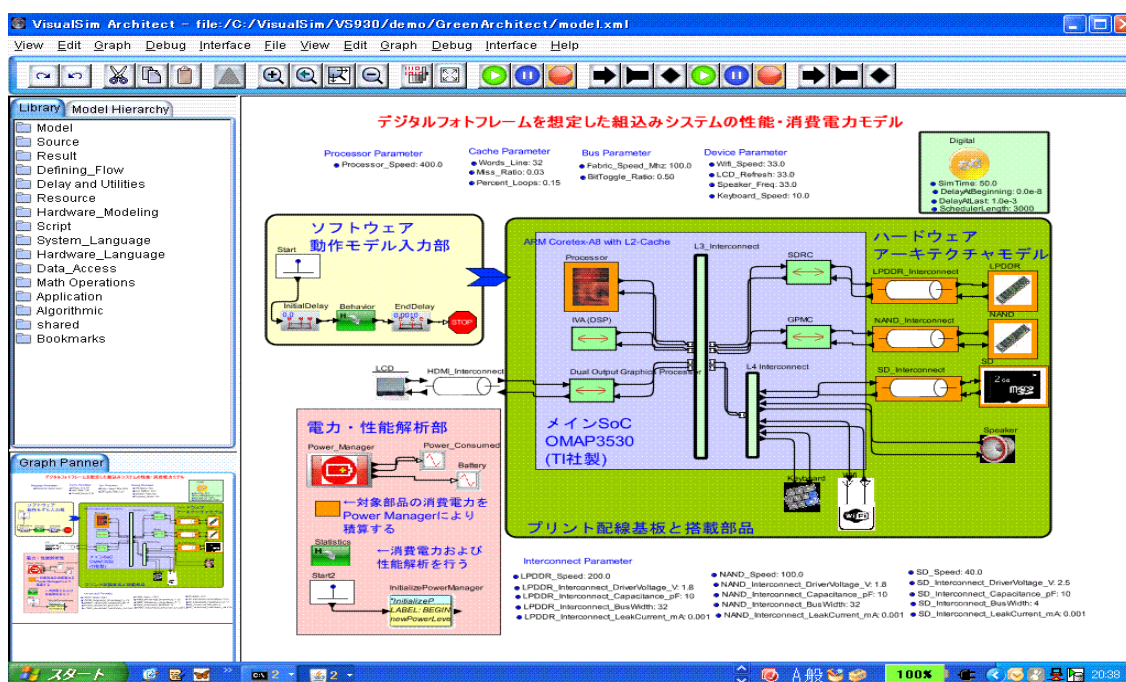


図2-2-5 ターゲット・システムのモデル構成例(シミュレータ:VisualSim)

### (3) ターゲット・システムの消費電力算出用のモデル構成

入力スティミュラスとしては、命令セットシミュレータからの出力として得られる各電子部品に対するアクセス統計情報を用いた。

各電子部品の消費電力は、各電子部品の状態に応じた消費電力値とその状態になる時間から算出した。一方、各伝送路の消費電力は、Read/Write のそれぞれについて消費電力値( $P_{\text{read}}$ ,  $P_{\text{write}}$ )を伝送路のパラメータ(参照)から算出した。

各電子部品の動作は、VisualSim の構成要素コンポーネントである「スケジューラ」を用いた。スケジューラは、各電子部品 (VisualSim 上の「リソース」) の動作状態をスケジューリング・アルゴリズムに基づいて表現できるコンポーネントである。これにより、各電子部品及び各伝送路が、使用される時間をそれぞれのスケジューリング・アルゴリズムに応じて模擬することにより、各電子部品及び各伝送路の動作状態をシミュレーションすることで、システム全体の性能と消費電力とをシミュレーションした。

表 2-2-2 システムレベルのモデル概要

| 項目              | 説明                                                                                       |                                                                                                                                                                                       |
|-----------------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 目的              | ターゲット・システムを構成する各電子部品と各伝送路の動作をシミュレーションし、消費電力を算出する                                         |                                                                                                                                                                                       |
| 入力スティミュラス       | マイクロプロセッサ(CPU)から各電子部品へのアクセス統計情報<br>・アクセス先(電子部品名)<br>・アクセス・タイプ (Read/Write)<br>・アクセス回数(回) |                                                                                                                                                                                       |
| 出力              | シミュレーション時間毎の各電子部品の消費電力値(mW)<br>(シミュレーション時間毎に変化する動的な値)                                    |                                                                                                                                                                                       |
| システムを構成するモデルの種類 | 各電子部品のモデル                                                                                | ・処理時間: nsec<br>・動作状態: Active/Standby                                                                                                                                                  |
|                 |                                                                                          | ・消費電力<br>Active 消費電力(mW)<br>Standby 消費電力(mW)                                                                                                                                          |
|                 | 各伝送路のモデル                                                                                 | ・動作状態: Read / Write / BurstRead / BurstWrite                                                                                                                                          |
|                 |                                                                                          | ・消費電力<br>Read 時の消費電力(mW): $P_{\text{Read}}$<br>Write 時の消費電力(mW): $P_{\text{Write}}$<br>BurstRead 時の消費電力(mW): $P_{\text{BurstRead}}$<br>BurstWrite 時の消費電力(mW): $P_{\text{BurstWrite}}$ |
| 電力の計算方法         | システム全体の消費電力 = 各電子部品の消費電力 + 各伝送路の消費電力(計算式は伝送路モデルの仕様を参照)                                   |                                                                                                                                                                                       |

### (4) システムレベルシミュレータ上の伝送路モデルの設計

伝送路モデルは、伝送路モデル仕様に基いて実装した。

表 2-2-3 各伝送路モデルの設計

| 項目        | 説明                                                                                                                                                                                                                                                                 |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 目的        | 伝送路の消費電力をユースケース(アプリケーション・ソフトウェアの動作)に応じて算出する                                                                                                                                                                                                                        |
| 伝送路モデルの駆動 | ・伝送路のアクセス・タイプ (Read/Write 等)<br>・アクセス回数(ターゲット・システムのビヘイビア・モデルからリソース(伝送路)へのマッピングによる。アプリケーション・ソフトウェアの動作に応じてシミュレーションされる。)<br>・ビット反転率(オプション)                                                                                                                            |
| 出力        | 伝送路の消費電力<br>※伝送路の処理時間(遅延時間)は、電子部品の処理時間に組み込まれているため、モデル化する必要がない。                                                                                                                                                                                                     |
| 設定パラメータ   | 伝送路の動作周波数: Interconnect_Speed (MHz)<br>伝送路のデータバス幅: Interconnect_BusWidth (bit)<br>伝送路の電気容量: Interconnect_Capacitance_pF (pF)<br>伝送路の I/O ドライバ電圧: Interconnect_DriverVoltage_V (V)<br>伝送路の動作に応じた計数パラメータ: ReadFactor, WriteFactor, ReadBurstFactor, WriteBurstFactor |

|                    |                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------|
|                    | 伝送路の動作に応じた計数パラメータ2:BurstLength(バースト長)                                                         |
| 伝送路の動作に応じた電力値の設定方法 | シミュレーション開始時に、PowerManager の API(powerUpdate)を使用して Read/Write/BurstRead/BurstWrite 状態の電力値を設定する |
| 電力の計算式             | 伝送路モデル仕様を参照                                                                                   |

#### (5) ターゲットシステムの処理性能

ターゲット・システムの処理性能(処理時間)をシミュレーションするために、各電子部品の性能モデルを設計した。各電子部品の処理時間は、ビヘイビア・モデル(動作モデル)の中で該当する電子部品による処理の遅延時間として扱った。

表 2-2-4 ターゲット・システムの処理性能のモデル化

| 項目            | 説明                                                                                        |
|---------------|-------------------------------------------------------------------------------------------|
| 目的            | ターゲット・システムの処理性能(処理時間)をシミュレーションする                                                          |
| 入力ステイミュラス     | マイクロプロセッサ(CPU)から各電子部品に対するアクセス統計情報<br>・アクセス先(電子部品名)<br>・アクセス・タイプ(Read/Write)<br>・アクセス回数(回) |
| 出力            | ターゲット・システムの動作タイミング<br>【図 8 に示す】                                                           |
| 処理時間(性能)の算出方法 | ターゲット・システム上の各電子部品のコンカレントな動作をシミュレーションし、その結果として得られる各部品の動作タイミングからシステム全体の遅延時間を計算する            |

ターゲット・システムは、以下の図 2-2-6 に示すように動作する。

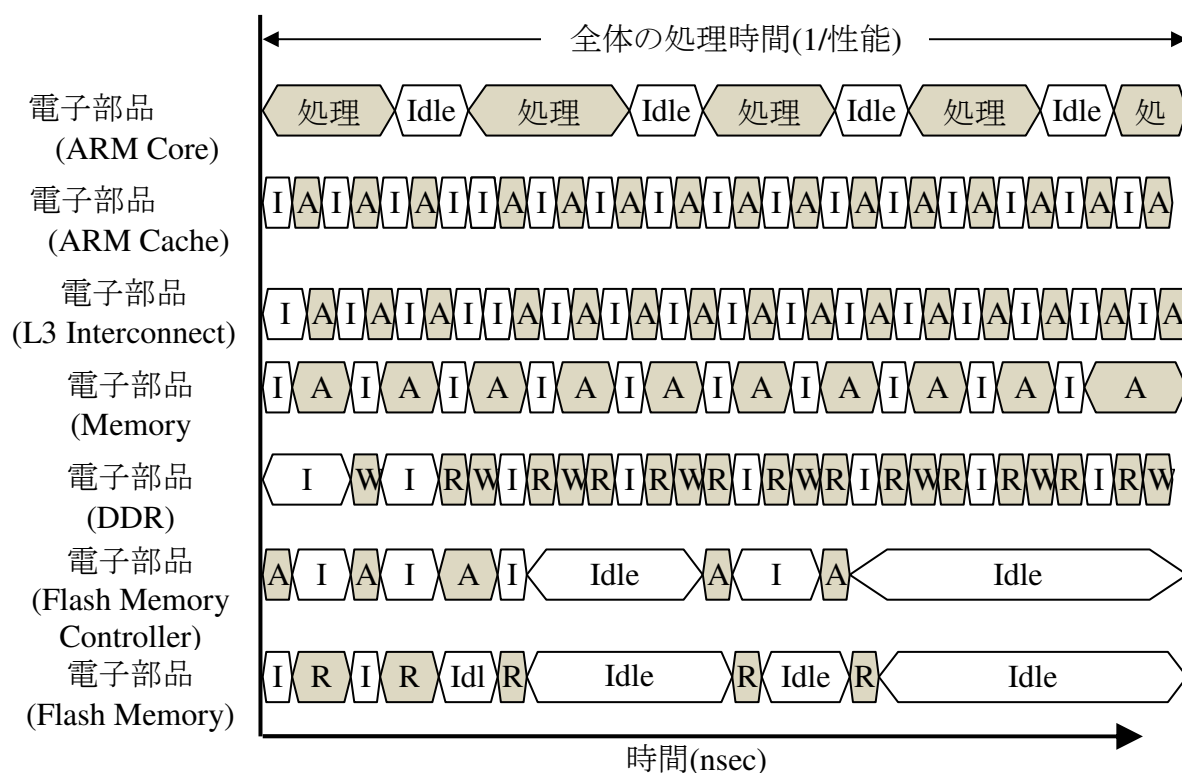


図 2-2-6 ターゲット・システムの性能(動作タイミング)

本研究開発では、シミュレーション速度を向上するため、ターゲット・システムの処理を図 2-2-7 に示すようにモデル化した。

まず、命令セットシミュレータ上の機械語命令列をある程度まとめて一つのタスクという単位とする。機械語命令列の長さは、10 万命令から 100 万命令とする。この機械語命令列を処理した際の

各電子部品へのアクセスを、「電子部品へのアクセス統計情報」としてまとめて、命令セットシミュレータから出力する。

システムレベルシミュレータは、「電子部品へのアクセス統計情報」を入力として、各電子部品の処理時間をシミュレーションし、以下の様にタスクに対する処理時間をシミュレーションした。このタスク処理時間の逆数がターゲット・システムの性能ということになる。

DDR メモリへのメモリアクセスは、Read と Write にアクセスのタイプを分けた。ARM Core から、ARM Cache・L3 interconnect・Memory Controller を経由して、対象の DDR メモリへのアクセスを行う動作モデルとした。

また、それ以外の電子部品へのアクセスは、共通のチップ内バス(L3 interconnect)を経由して、各電子部品(Flash Memory)に、順次アクセスする動作モデルとした。

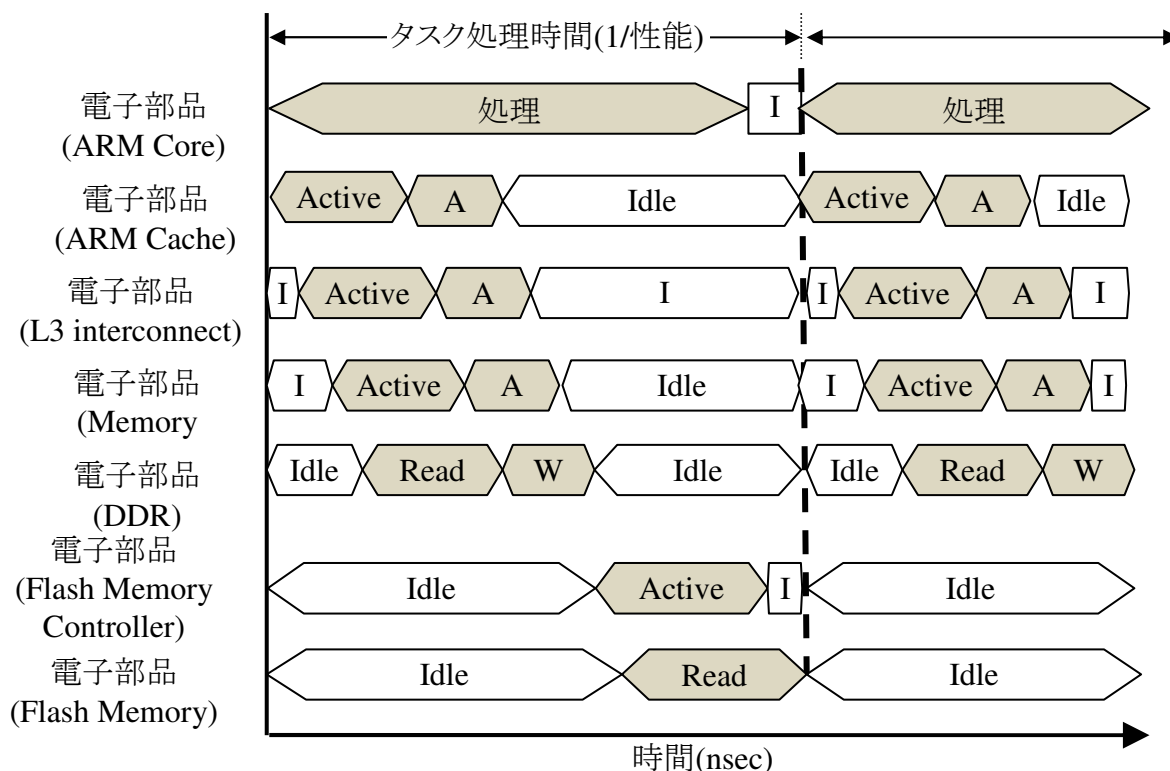


図 2-2-7 ターゲット・システムの性能(動作タイミング)

図 2-2-8 に、本研究開発において構築した、性能シミュレーションのための動作モデルの画面写真を示す。

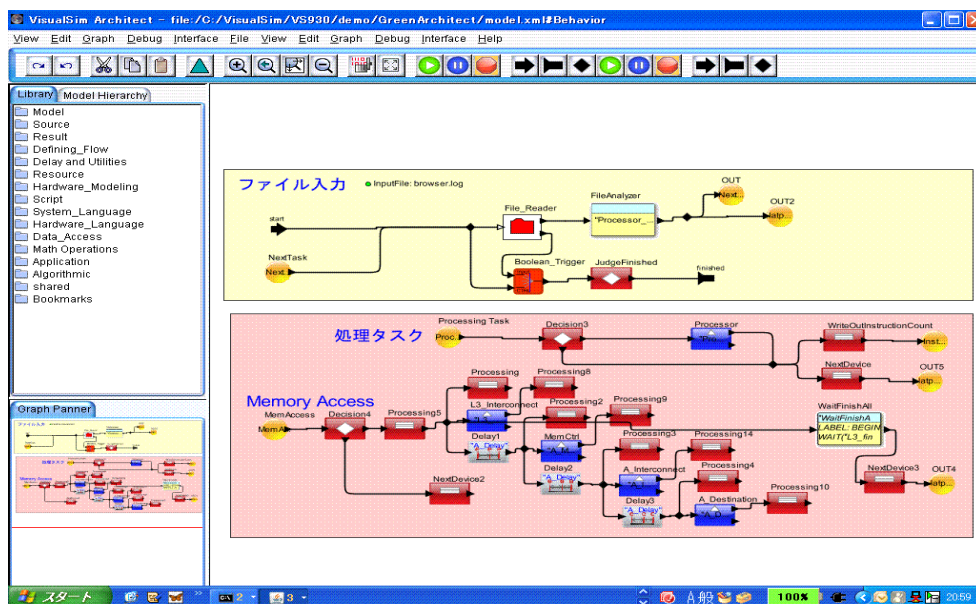


図 2-2-8 ターゲット・システムの動作モデル (VisualSim)

#### (6) 命令セットシミュレータとハードウェア・モデルの統合手法

本研究開発では、アプリケーション・ソフトウェアを動作させ、ターゲット・システムの消費電力シミュレーション結果をインタラクティブに観測できるようにするために、命令セットシミュレータ(ISS)とシステムレベルのシミュレータ(VisualSim)を統合した。

命令セットシミュレータは、CPU の動作を機械語命令単位で模擬することにより、OS を含むアプリケーション・ソフトウェアをそのまま実行できる。また、本研究開発で使用した QEmu のように命令セットシミュレータに電子部品の動作モデル(Goldfish)を組込むことにより、CPU だけでなく、周辺メモリや周辺デバイスを含む動作をシミュレーションすることも可能。但し、電子部品の処理時間はモデル化されない。

従って、システムの処理性能をシミュレーションするには、命令セットシミュレータとシステムレベルシミュレータ上の電子部品のモデルを統合する必要がある。

命令セットシミュレータとシステムレベルシミュレータとの統合方法について、①ファイルを経由する統合、②プロセス間通信による統合、③命令セットシミュレータ(ISS)とハードウェア・モデルの統合の 3 種類が考えられる。それぞれの長所短所について以下に示す。

表 2-2-5 命令セットシミュレータとシステムレベルシミュレータとの統合方法

|                       | ファイル経由する統合  | プロセス間通信による統合<br>(CORBA 等) | ISS とハードウェアのモデルを<br>統合 (SystemC 等) |
|-----------------------|-------------|---------------------------|------------------------------------|
| 結合度                   | 疎           | やや疎                       | 密                                  |
| 開発期間                  | 短い          | やや短い                      | 長い                                 |
| ソフトウェアと<br>ハードウェア     | 独立にシミュレーション | 同時にシミュレーション               | 同時にシミュレーション                        |
| シミュレーション<br>速度        | 速い          | やや遅い                      | やや遅い                               |
| ソフトウェアと<br>ハードウェアの同期  | 不可          | 可能                        | 可能                                 |
| ソフトウェアと<br>ハードウェアの同期の | 0.1～1M 命令程度 | 0.1～1M 命令程度               | バス・サイクル                            |

|                |                           |                           |   |
|----------------|---------------------------|---------------------------|---|
| 時間的粒度          |                           |                           |   |
| シミュレーション<br>精度 | △<br>各電子部品とCPUの同期待ちを考慮しない | △<br>各電子部品とCPUの同期待ちを考慮しない | ○ |

本研究開発ではシミュレーションの速度を優先し、①ファイルを経由した接続の方法を採用した。

シミュレーション速度向上の観点から、命令セットシミュレータから出力するアクセス統計情報は、CPUが10万命令もしくは100万命令毎とした。命令シミュレータから出力するアクセス統計情報量を大幅に削減することで、シミュレーション速度を大幅に向上させることができた。

## 2-3 アプリケーションソフト動作時のシステム電力評価

### (1) 目的

ターゲット・システム上でアプリケーション・ソフトウェアを実行した時の消費電力を測定し、ターゲット・システムの性能評価用モデルによる電力シミュレーション結果と比較し、各電子部品及び各伝送路モデルのキャリブレーションを行うことを目的とした。

### (2) 対象

消費電力シミュレーションの対象とするアプリケーション・ソフトウェアは、携帯情報端末の代表的なアプリケーションである WEB ブラウザと動画プレーヤを選択した。

シミュレーションに先立ち、命令セットシミュレータを用いて WEB ブラウザ・動画プレーヤを動作させた際の、アクセス統計情報を取得した(表 2-3-1)。

比較として、その他のアプリ(GoogleMap)、GUI 操作等の実行時のアクセス統計情報を示す。

表 2-3-1 アプリケーション動作統計情報データ概要

|                  | WEBブラウザ       | 動画プレーヤ        | その他アプリ<br>(GoogleMap) | GUI操作等        |
|------------------|---------------|---------------|-----------------------|---------------|
| InstructionCount | 2,713,609,309 | 2,713,958,686 | 1,995,126,974         | 1,029,248,080 |
| DDR_Read         | 23,476,859    | 25,268,162    | 22,134,535            | 12,916,465    |
| DDR_Write        | 31,647,297    | 42,029,924    | 30,700,860            | 17,788,575    |
| NAND_Read        | 4,617,804     | 319,808       | 1,311,160             | 3,690,868     |
| NAND_Write       | 957,036       | 26,988        | 276,108               | 325,932       |
| SD_Read          | 0             | 2,803,332     | 2,060                 | 24            |
| SD_Write         | 0             | 3,766         | 20                    | 2,596         |
| Wifi_Recv        | 125,171       | 0             | 924,071               | 0             |
| Wifi_Send        | 20,520        | 0             | 46,278                | 0             |

メモリ別のメモリアクセス回数を示したのが図 2-3-1 である。動画プレーヤ実行時は、Web ブラウザ実行時と比べて DDR メモリへのアクセスの割合が多く、SD メモリへのリードアクセスが多いことが分かる。また、NAND フラッシュメモリへのアクセスは、Web ブラウザの方が多い。これらのアクセス割合の違いとアクセス量の違いが、システム全体の消費電力の違いとなることが予測できる。

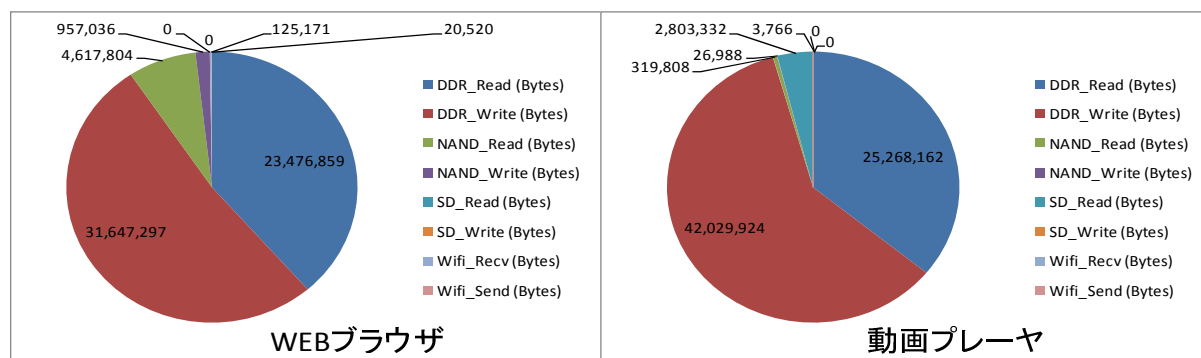


図 2-3-1 アプリケーション動作時の対象メモリ別メモリアクセス量分析

### (3) WEB ブラウザ動作時のシステム電力評価結果

WEB ブラウザ実行時のアクセス統計情報に基づき、システム全体の消費電力をシミュレーションした結果を以下に示す。

システムの構成： ARM プロセッサ、L3Interconnect、メモリコントローラ、DDR メモリ、フラッシュメモリコントローラ、NAND フラッシュメモリ、SD メモリコントローラ、SD メモリカード

表 2-3-2 WEB ブラウザ動作時のシステム電力シミュレーション結果

|               | シミュレーション結果                       |
|---------------|----------------------------------|
| システム全体の平均消費電力 | 1182mW                           |
| 処理時間          | 7.83sec                          |
| シミュレーション速度    | 7.45M 命令/秒<br>(2713M 命令 = 364 秒) |

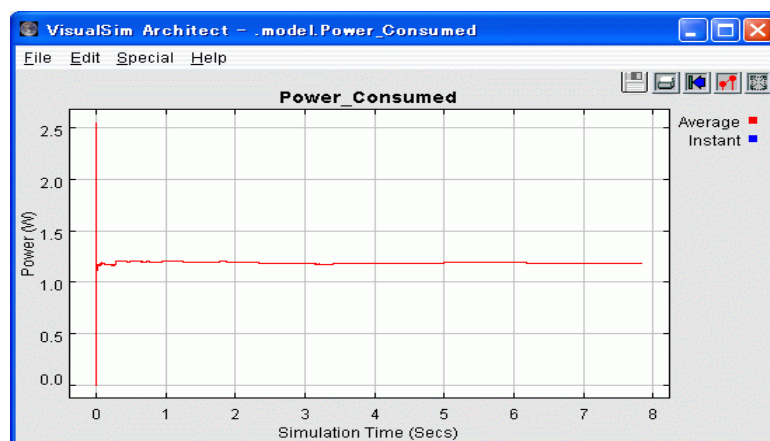


図 2-3-2 WEB ブラウザの消費電力見積もりシミュレーション結果

[伝送路の消費電力について]

システム全体の消費電力に対して、伝送路の消費電力の占める割合は決して少なくはない。例えば OMAP3530 のデータシートによると、OMAP3530 のコア電力の合計は最大 2.3W 程度であるのに対して、I/O 電力の合計は最大 470mW 程度である(メモリコントローラ・ドライバ含む)。すなわち、伝送路電力はシステム電力の約 20%に相当する。

より詳細な伝送路の影響について調べるため、伝送路モデルのパラメータを変化させた際のシステム全体の消費電力を調べた。

入力として、WEB ブラウザ動作時とシステム起動時のアクセス統計情報からそれぞれ一部を切り出したものを用いた。

表 2-3-2 に、電力評価を行うために用いた、各電子部品に対するアクセス統計情報と、その特徴を示す。

表 2-3-2 伝送路の消費電力評価に用いたアクセス統計情報

| 各電子部品に対するアクセス統計情報 | 特徴             |
|-------------------|----------------|
| システム起動時1 (Boot1)  | メモリアクセスが比較的少ない |
| システム起動時2 (Boot2)  | メモリアクセスが中程度    |
| Web ブラウザ(browser) | メモリアクセスが比較的多い  |

図 2-3-3 は、ARM コア、L3 Interconnect、メモリコントローラと DDR メモリとで構成されるシステムについての、伝送路の違いによるシステム全体の消費電力シミュレーション結果の比較である。伝送路は、データ幅 32bit の LPDDR メモリとのインターフェイスである。この伝送路の 1 ビット当たりの負荷容量を、10pF・20pF・30pF と変化させた場合に、システム全体の消費電力値が増加することが分かる。

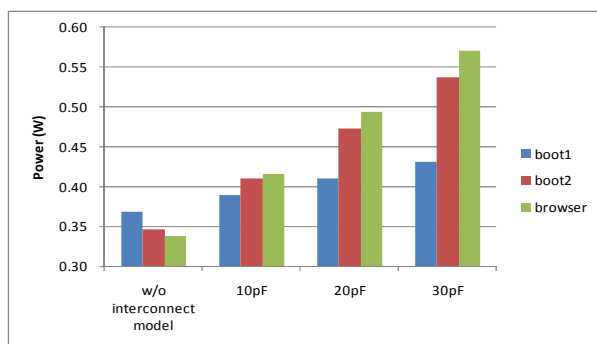


図2-3-3 伝送路の違いによるシステム全体の消費電力(シミュレーション結果)

また、図 2-3-4 に、伝送路のデータビット幅の違いシステムの性能(処理時間)とシステムの消費電力のシミュレーション結果を示す。伝送路である LPDDR メモリとのインターフェイスのビット幅を増加させると、一度のアクセスで駆動される配線の本数が増加するために 1 回のアクセスで消費される電力が大きくなるのでシステムの消費電力は増加する。一方、一回のアクセスに要する処理時間は変わらないが、アクセス回数が減少するため、アプリケーションの処理時間は減少する。シミュレーション結果から、この傾向が観測できた。

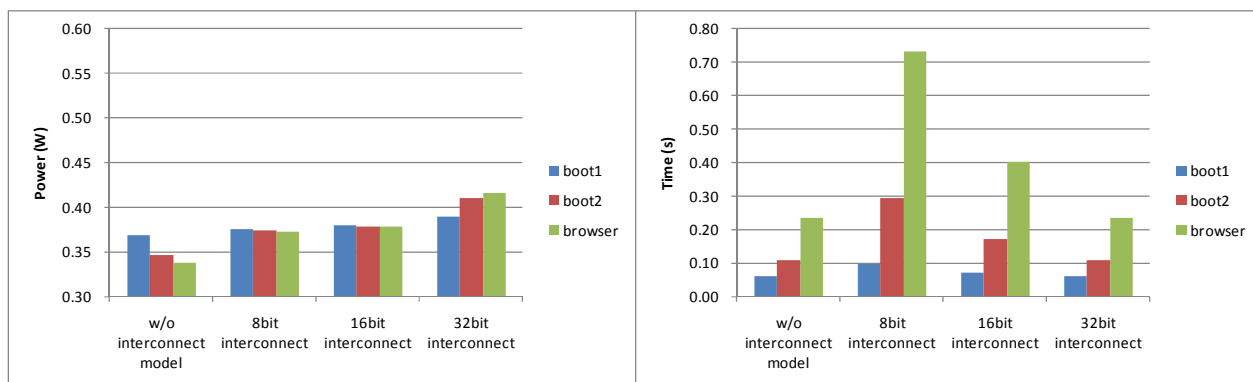


図2-3-4 伝送路のデータビット幅の違いによる及び消費電力及び性能シミュレーション結果

#### (4) 動画再生アプリケーション動作時のシステム電力評価結果

動画再生アプリケーション実行時のアクセス統計情報を使用して、伝送路を含むシステム全体の消費電力を評価した結果を示す。

システムの構成: ARM プロセッサ、L3Interconnect、メモリコントローラ、DDR メモリ、フラッシュメモリコントローラ、NAND フラッシュメモリ、SD メモリコントローラ、SD メモリカード

表 2-3-3 動画再生アプリケーション動作時のシステム電力シミュレーション結果

|               | シミュレーション結果                       |
|---------------|----------------------------------|
| システム全体の平均消費電力 | 1331mW                           |
| 処理時間          | 7.21sec                          |
| シミュレーション速度    | 7.13M 命令/秒<br>(2713M 命令 = 380 秒) |

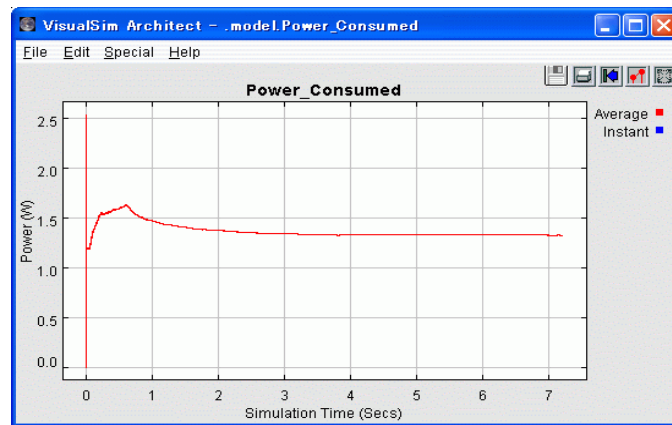


図 2-3-5 動画像再生アプリケーション実行時のシステム全体の消費電力(シミュレーション結果)

#### (5) システムの消費電力評価方法

アプリケーション動作時のアクセス情報統計の出力には、トプスシステムズ社製の Android On VisualSim に内蔵する命令セットシミュレータ QEmu(オープンソース)を改造し、CPU から電子部品へのアクセス統計情報を出力可能にした。以下に、出力した電子部品へのアクセス統計情報の項目を示す。

- 実行した命令数 (Instruction Count)
- DDR\_Read/Write 回数
- Flash\_Read/Write 回数
- SD\_Read/Write 回数
- FB\_Write 回数 (フレームバッファ: LCD 表示用)
- Wifi\_Send/Recv 回数 (無線 LAN)

また、電子部品へのアクセス統計情報の例を図 2-3-6 に示す。

```
{ InstructionCount = 1028157, InstructionCycle = 1943445, ICacheHit = 9662, ICacheMiss = 1285, DCacheReadHit = 294908, DCacheReadMiss = 14365, DCacheWriteHit = 187583, DCacheWriteMiss = 3538, DDR_Read = 15650, DDR_Write = 17903, NAND_Read = 4152, NAND_Write = 8304, SD_Read = 0, SD_Write = 0, FB_Write = 393216000, Wifi_Send = 0, Wifi_Recv = 0, Index = 1491}
{ InstructionCount = 1023155, InstructionCycle = 1841315, ICacheHit = 7364, ICacheMiss = 833, DCacheReadHit = 300481, DCacheReadMiss = 13346, DCacheWriteHit = 168715, DCacheWriteMiss = 2015, DDR_Read = 14179, DDR_Write = 15361, NAND_Read = 0, NAND_Write = 0, SD_Read = 0, SD_Write = 0, FB_Write = 294912000, Wifi_Send = 0, Wifi_Recv = 0, Index = 1492}
...
```

図 2-3-6 電子部品へのアクセス統計情報の例

## 2-4 アーキテクチャ探索システムの実装組合せ制約条件に基づく網羅方法の検討

### (1) 目的

電子部品を組み合わせてシステムを構成する際には、膨大な組み合わせの可能性がある。

システムを構成する膨大な部品は、1つの部品を取ってみてもメーカー、金額、性能(ただし最低限の機能は同じ)、形状、大きさはさまざまである。よってシステムを作成するために開発者が選択しなければならない部品の組み合わせは無数になってしまう。

本研究開発のアーキテクチャ探索システムは、それらの膨大な組み合わせの中から、アプリケーションに応じて消費電力・性能を最適化する接続方法、すなわちアーキテクチャを探索することを目的とした。

### (2) 検討内容

本研究開発においては、対象のハードシステムは携帯情報端末とした。さらに携帯情報端末に使う部品のうち、特に消費電力に影響すると思われる部品にのみに着目した。

部品とは、抵抗、コンデンサのような単機能の物ではなく、システムの機能の一部として定義できるレベルのものである。それらの部品についての組み合わせを検討できるようにした。

今回の開発で部品として扱ったものを以下に示す。

- ・ CPU チップ
- ・ LDC
- ・ メモリ(FLASH メモリ、DDR メモリ、SD カード)
- ・ WiFi チップ
- ・ バス
- ・ 伝送路モデル(CPU とメモリ間)

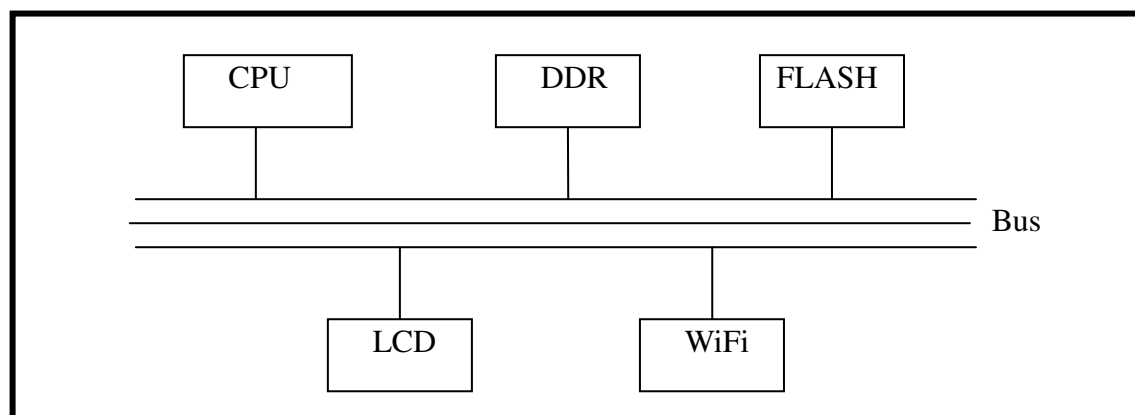


図2-4-1 システム構成例

これらの部品について、それぞれ何種類かの候補があった場合にすべての組み合わせを単純に行うと、かなりの組み合わせになってしまう(例としてそれぞれ2候補あった場合でも $2^6=64$ 通り)。

下記、表 2-4-1、表 2-4-2 参照

表2-4-1 各部品の候補がそれぞれ2 パターンの場合部品表

| 部品タイプ | 部品候補-1 | 部品候補-2 |
|-------|--------|--------|
| CPU   | ARM9   | ARM11  |
| DDR   | SMEM1  | SMEM2  |
| ...   | ..     | ..     |
| WiFi  | WCHP1  | WCHP2  |

表2-4-2 各部品の候補を組み合わせた場合のシステムパターン表

| システム構成   | CUP   | DDR   | ... | WiFi  |
|----------|-------|-------|-----|-------|
| System1  | ARM9  | SEME1 | ..  | WCHP1 |
| System2  | ARM9  | SEME1 | ..  | WCHP2 |
| System3  | ARM9  | SEME1 | ..  | WCHP1 |
| ..       | ..    | ..    | ..  | ..    |
| SystemN  | ARM11 | SEME2 | ..  | WCHP1 |
| ..       | ..    | ..    | ..  | ..    |
| System64 | ARM11 | SEME2 | ..  | WCHP2 |

システムとしてはそれらのすべてについてシミュレーションを実行して、その中から最適なシステム構成を抽出することは大きな問題ではないが、その中には不要な組み合わせが存在している場合が多々ある。たとえば、「CPU-A は 32 ビットなので、到底8ビットバスは使えない」とか、「システム全体の大きさから考えて、この LCD とこのメモリは同時には使えない」とか...である。

このような制約が現実的には存在するため、本システムでは、部品の組み合わせにおいてある制約を与えることで、不要な組み合わせのシミュレーション実行を予め省くことが出来るように考慮した。このことによって実際に最適システム探索に要する時間を大幅に短縮することが可能となった。

制約の与え方は個々の部品候補ごとに設定できるようにした。以下に部品毎の制約例を示す。

表2-4-3 部品候補間の制約例

| 部品候補名A | 部品候補名B | 制約        |
|--------|--------|-----------|
| ARM9   | AMBUS8 | NM (混載不可) |
| ARM11  | SMEM3  | P (必ずペア)  |
| LSI100 | LSI300 | NM        |
| LSI100 | SD1    | P         |
| ..     | ..     | ..        |

制約の種類としては、2 種類を考慮した。

混載を許さない指定 (NM) と、必ずペアで使う指定 (P) である。

これらの指定が候補対象の部品に指定されていた場合は、組み合わせパターンについてチェックを行い、制約を守ったシステム構成についてのみシミュレーションを実行するようにした。さらに、部品間の伝送路モデルも複数準備し、これらの組み合わせによるシステムの消費電力比較も行えるようにした。

伝送路モデルとしては、

- (a) IP を SoC 内での集積
- (b) LSI チップとしてプリント配線基板上での集積
- (c) SIP としてワイヤボンディングによる集積

#### (d) 3次元積層 LSIとして貫通電極(TSV)で集積

などについてモデル化し、他の部品と同じようにシミュレーション毎に伝送路モデルを入れ替えて消費電力をシミュレーションできるようにした。

伝送路にまで消費電力モデルを適応した理由は、電子部品間のインターフェイスに消費される電力は、LSI チップの全消費電力の 10-30%と小さいが、チップ実装方法により配線容量は 100 倍以上になることがあると言われているからである。

伝送路モデル定義に際しては、3 種類の異なる伝送路形態の測定カスタムボード作成を行い、実測することで、伝送路モデルへ反映した。当然、伝送路モデルと一般部品との間においても同上の制約を設定することは可能である。

本システムでは、今回対象としている携帯情報端末について、どのアプリケーション動作時のものを対象にシミュレーションするかで、その実行時間は大きく違ってくる。また携帯情報端末に限らず任意のシステム(測定対象のシステム)のシミュレーションを行う場面において、今後はその実行回数が増えることが容易に、予想できる。(いろいろな環境下を想定して、シミュレーション実行を行いたいからである)

このようなことを考えると、単純な総当りでのシミュレーション実行ではますます実行時間が長くなってきてしまい、シミュレーションしたい環境を減らすなどの対策をしなければいけない可能性も出てくる。このような状況においては、非現実的なシミュレーション環境を想定した実行を出来るだけ少なくする必要がある。そのために「制約」を与えることは有益なことであり、かつその制約を自動的に解釈し、不要なシミュレーション実行を省くことが出来ることはシミュレーション実行環境としては大事な一機能である。

## 2-5 アーキテクチャ探索システムの開発

### (1) 目的

アーキテクチャ探索システムは、製品ターゲットのシステム（本研究開発では携帯情報端末を想定）を構成する部品について、どの組み合わせがもっとも製品要求にマッチしているかをシミュレーション実行し、その結果や、その他の付加情報を使って導き出すことが出来るソフトウェア環境を構築することを目的とした。

### (2) 開発内容

本システムでは、消費電力をシミュレーションするために **VisualSim** というソフトウェアを使った。本システムはこの **VisualSim** の実行制御と結果の検索が出来るような環境作成を行った。

**VisualSim** は消費電力のシミュレーションだけでなく、機能、動作などのシミュレーションを行うことが可能であるが、本開発では消費電力に着目しているため、他のシミュレーション実行は行わなかった。

以下が本システムの構成図である。

オペレータからの指示は専用の **GUI** にて処理され、マニュアルなどの熟読無しに操作ができるように、直感的に使えるよう工夫をしている。ユーザからの指示が **GUI** に渡されると、指示内容のチェックがされ、不適切な指定に対しては再入力を促すようになっている。

**GUI** 上で指定内容のチェックが行われ、問題が無ければ各制御部にその指示が渡され、指示にあった処理が動く。

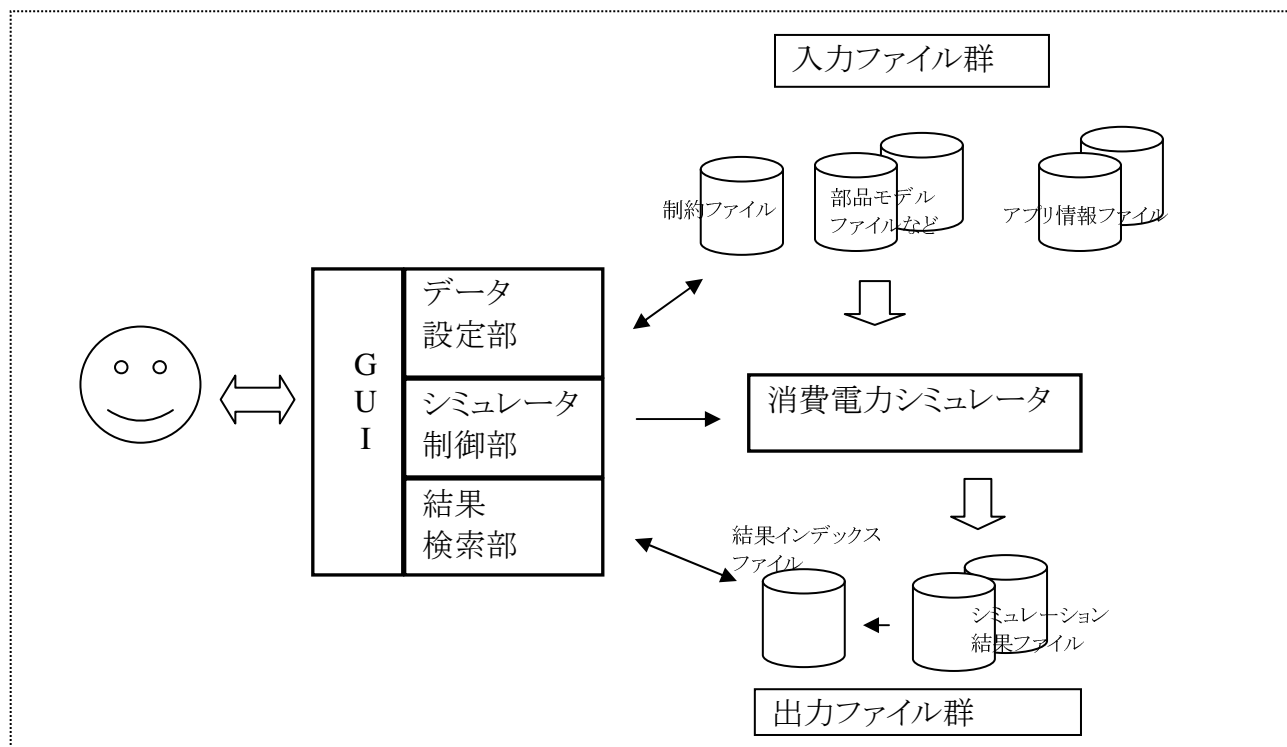


図2-5-1 アーキテクチャ探索システムソフトウェア構成図

本システムの主な機能は以下の通りである。

- ・ データ設定機能
- ・ VisualSim（消費電力シミュレータ）の実行制御機能
- ・ VisualSim 結果の検索機能

以下にそれぞれの機能について説明する。

(a) データ設定機能

データ設定機能とは、消費電力測定対象システムの構成を変更するために必要な情報を解析し、シミュレータ入力用のファイルを加工、生成する機能である。

たとえば、バスのビット幅を 16 ビットから 32 ビットに変更して測定する場合や、異なるメーカーのメモリを使って消費電力測定比較を行いたい場合などである。このような指定がされた場合、対応するパラメータ値を変更した入力ファイルの作成や、メモリ部品の定義部分の入れ替えを行った入力ファイルの作成を行う。パラメータ値の変更が 2 回、部品の変更が 2 回指定された場合でも、単純に 4 パターン（2 x 2）の入力ファイルの作成を行うわけではなく、制約ファイルに部品間の制約情報が記載されていれば、その条件を満たす組み合わせのみの入力ファイル作成を行う（制約違反になる部品の組み合わせは行わない）。これによって総当り組み合わせによる無駄なシミュレーション実行をなくすことが出来るようになっている。

以下に実際の制約ファイルの例を示す。

ファイルのフォーマットとしては、**\$MUSTLINK** キーワードから次のシステムキーワードまでの間に、1 行ごと 2 ワード（2 部品）の記述が出来、その 2 ワード間にはどちらかの部品が使われている場合、もう一方の部品と一緒に使われなければならないという指定ができる。さらに **\$NGLINK** キーワードから次のシステムキーワードまでの間では、構成要素として一緒に使ってはいけない部品の組み合わせが指定できる。

例では、**CUP1** と **MEM100** は同時に構成されなければならない、**PARTN** と **PARTM** も同時に構成されなければならない指定になる。もしオペレータからの指定がこれに矛盾する場合は、その組み合わせでの実行は自動的に行わなくなる。

さらに **PART10** と **PARTX** については同時にシステムを構成してはいけない指定になっているため、オペレータからの指定で **PART10** と **PARTX** を同時に使ったシステム構成でのシミュレーション実行指定がされたとしても、実行は自動的に行われなくなる。

```
.....  
$MUSTLINK  
"CPU1" "MEM100"  
"PARTN" "PARTM"  
  
$NGLINK  
"PART10" "PARTX"  
.....
```

なお上記ファイルには、部品の制約情報以外の付加情報も記載できるため、前後にはその他のセクションが記述される。

以下に、GUI にてデータを設定するダイアログを示す。

図 2-5-1 のネットリスト&パラメータ指定ダイアログでは、システム構成の接続関係を表しているネットリストファイルを指定する。またそのネットリスト内で変更し

でシミュレーションを実行したいパラメータがあれば、そのパラメータを指定することも可能である。変更したいパラメータは複数指定可能である。

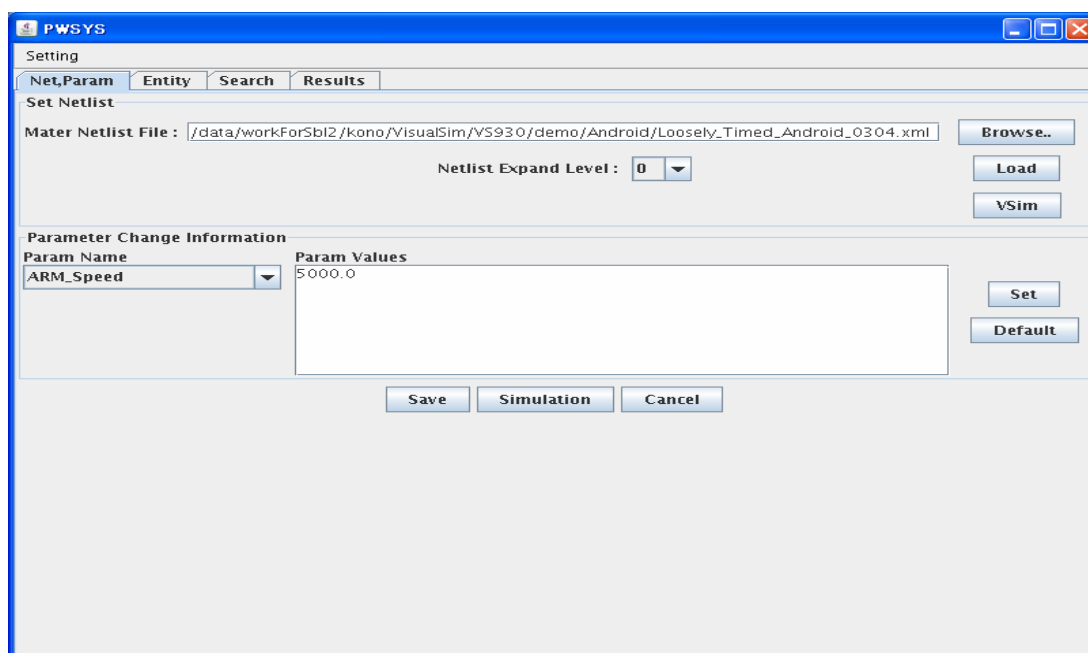


図 2-5-1 ネットリスト&パラメータ指定ダイアログ

図 2-5-2 の部品入れ替え指定&シミュレーション実行指定ダイアログでは、図 2-5-1 のパラメータ変更とは別に、部品ごと変更してシミュレーションを実行したい場合の指定が出来るダイアログである。パラメータと部品の同時変更も可能である。

部品の変更時には部品の制約指定に従って、不要な組み合わせはシミュレーション実行されないように制御される。

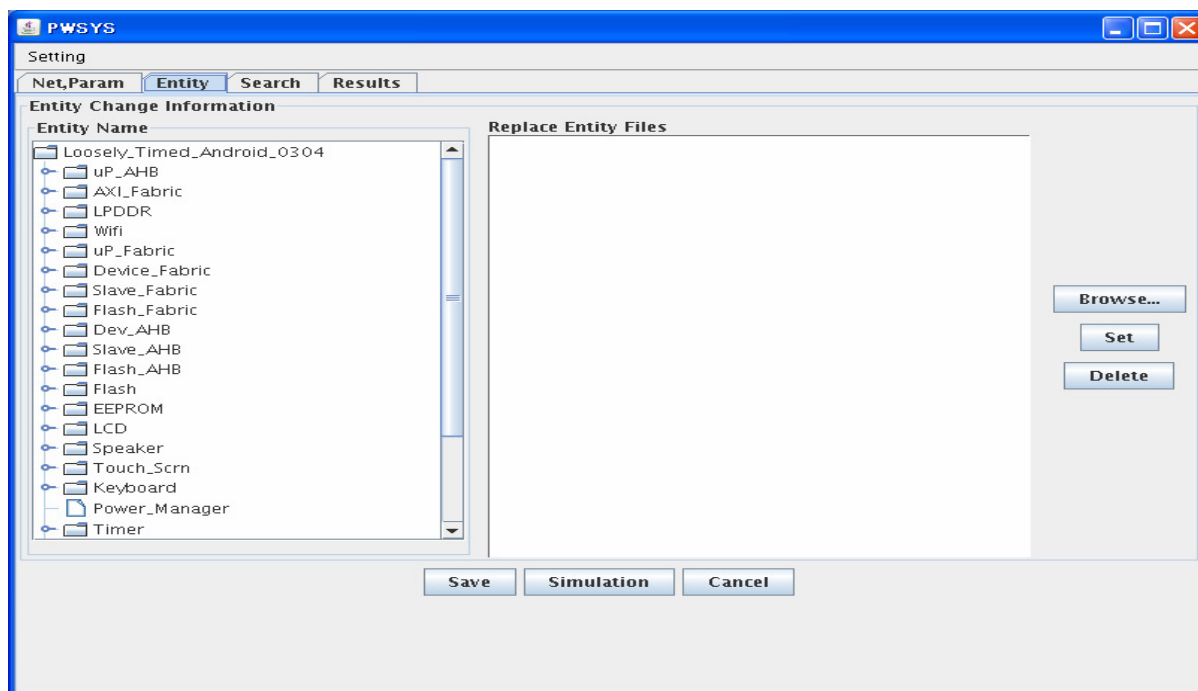


図 2-5-2 部品入れ替え指定&シミュレーション実行指定ダイアログ

## (b) VisualSim（消費電力シミュレータ）の実行制御機能

シミュレータの実行制御機能とは、データ設定機能により作成されたシミュレーション用の入力ファイルを使って、シミュレーションの連続実行が出来るようなバッチスクリプトファイルを作成する。システムとしてはこのバッチファイルを作成し、実行させることでシミュレータを起動することが出来る。

VisualSim を実行できる Linux マシンがネットワーク上に複数存在し、複数実行のライセンスを使えるのであれば、ライセンス分の実行を別々のマシンで実行し、より実行時間を短縮するような設定も可能である。

シミュレーション実行が終わると、実行結果の解析を行い、後の検索処理を高速化するためのインデックスファイルを作成する。このファイルを使うことでシミュレーション結果が膨大になった時の検索時間の短縮が可能になる。また特別なデータベースを作成して結果の管理を行う必要も無いため、開発効率の短縮にもつながった。

実際のシミュレーション実行については(a)データ設定機能内の GUI にて、データの設定後に起動が出来るようになっている。

シミュレーション実行が終了した後は、VisualSim の結果ファイルの解析が自動的に行われ、シミュレーション実行時の変更パラメータ情報や、変更部品情報、結果のサマリーなどがインデックスファイルに格納される。インデックスファイルは検索時のデータベースとして使用される。

## (c) VisualSim 結果の検索機能

VisualSim を使って行われた消費電力シミュレーション結果群は、専用の GUI を解して結果を検索することが出来る。

以下が専用の GUI である。（図 2-5-3 検索指定ダイアログ）

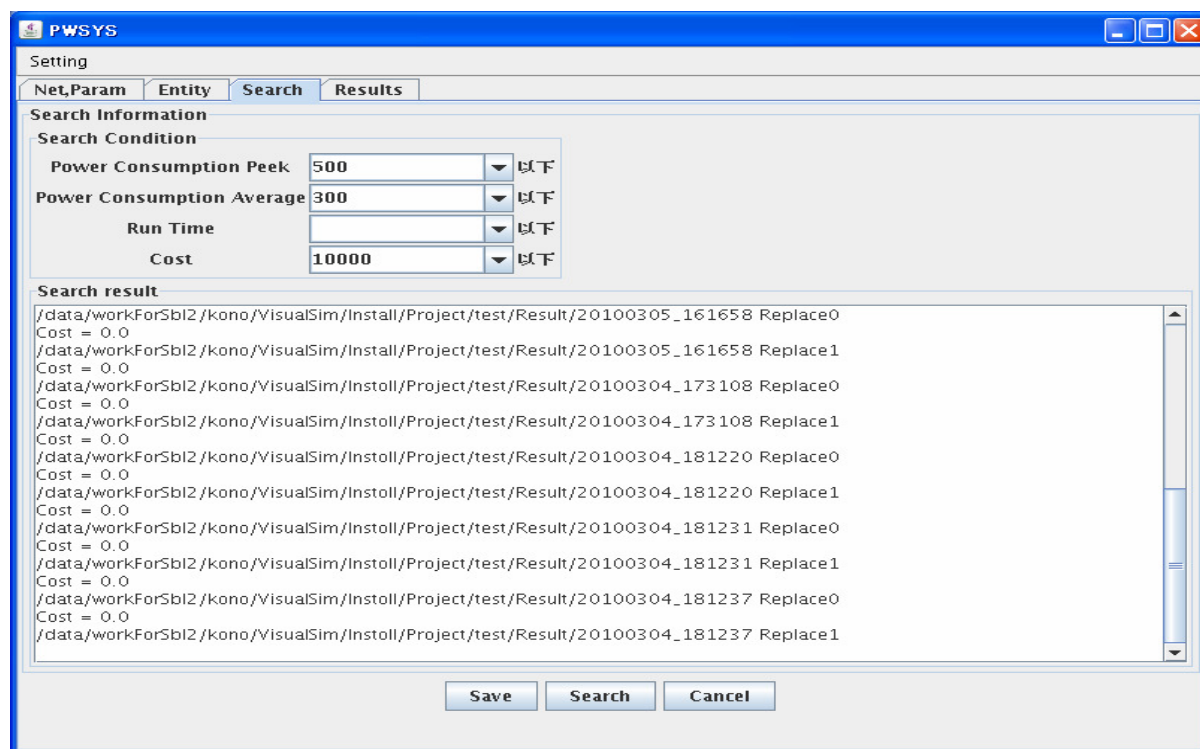


図 2-5-3 検索指定ダイアログ

検索指定の条件としては以下を考慮した。

- 消費電力の最大値

- ・ 消費電力の平均値
- ・ アプリケーションの実行時間
- ・ 部品の調達コスト

以上の指定項目で、複数指定された場合はそれらをすべて満足するシステム構成のものが検索される。検索結果は複数見つかる場合もある。検索結果の詳細を見る場合は結果一覧の中より1つを選択し、**Information** ボタンをクリックすることで、以下のようにシステム構成の詳細情報(図 2-5-5 システム構成詳細情報)を見ることが出来る。

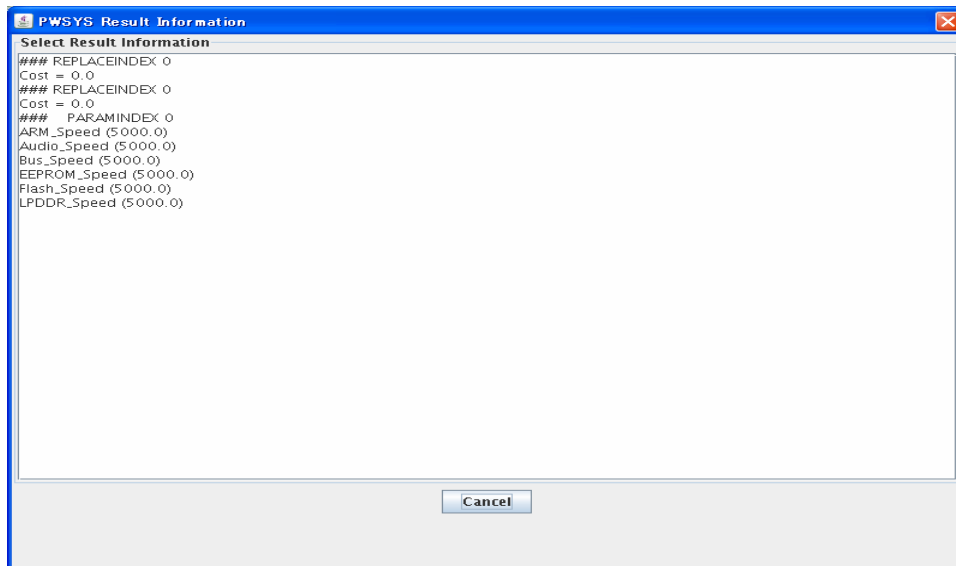


図 2-5-5 システム構成詳細情報

## 2-6 アーキテクチャ探索システムのテスト・検証

### (1) システムの機能テストおよび検証結果

開発したシステムの機能テストおよび検証を行った。

システムの機能テストおよび検証は、プログラムチェックリスト(PCL)を作成し、それぞれのチェック項目について実行し、所望の結果が出ているのかを確認した。

プログラムチェックリストの一般的な内容は、

- ・ AA を BB した時には、XX が YY になること。
- ・ AA を CC した時には、ZZ が WW にならないこと。

というように、確認したい動作について具体的なオペレーションと、期待値を明記した。これにより本システムの詳細な機能を理解していないオペレータでもテストの判定について○×をつけることが出来るようにした。結果の検証については基本的には目視で行った。ただしシミュレーション結果についての検証は基準となるシミュレーション結果をいくつかピックアップし、その結果を十分に検証し、その結果との比較において妥当性を確認することで、その他多くの検証処理の簡略化を計り、チェック時間の短縮も行った。

以下にいくつかのプログラムチェックの例を示す。今回のシステムの機能テストでは、以下のような項目を2000件弱準備し、テストを行った。

表 2-6-1 プログラムチェックリストの例

| 項目         | チェック条件                                                  | チェック内容                                       |
|------------|---------------------------------------------------------|----------------------------------------------|
| シミュレーション実行 | 伝送路モデルを1で実行する                                           | 伝送路モデルパラメータを1に変更してシミュレーションが正しく出来ること          |
|            | 伝送路モデルを2で実行する                                           | 伝送路モデルパラメータを2に変更してシミュレーションが正しく出来ること          |
|            | 伝送路モデルを3で実行する                                           | 伝送路モデルパラメータを3に変更してシミュレーションが正しく出来ること          |
|            | .....                                                   | .....                                        |
| データ格納      | 伝送路モデルを3、ビット幅を 32 ビット、負荷容量を 5u、周波数を 10Hz で実行し、出力結果を格納する | 出力ファイルが保存されていること                             |
|            | 伝送路モデルを3、ビット幅を 32 ビット、負荷容量を 5u、周波数を 20Hz で実行し、出力結果を格納する | 出力ファイルが保存されていること                             |
|            | 伝送路モデルを3、ビット幅を 32 ビット、負荷容量を 10u、周波数を 1Hz で実行し、出力結果を格納する | 出力ファイルが保存されていること                             |
|            | .....                                                   | .....                                        |
|            | 負荷容量の範囲を小数点数から整数で指定して入力して実行する                           | 範囲を区切り、区切られた値に全て負荷容量パラメータにそれぞれ入力してシミュレーションする |
|            | 負荷容量の範囲を小数点数から小数点数で指定して入力して実行する                         | 範囲を区切り、区切られた値に全て負荷容量パラメータにそれぞれ入力してシミュレーションする |

### (2) アーキテクチャの探索性能

性能テストを行いアーキテクチャの探索性能を評価した。

- ・部品数 : 10部品 + 50伝送路
- ・シミュレーション速度 : 1M 命令/秒

- ・アーキテクチャ探索時間: 1アーキテクチャあたり6000秒  
(100アーキテクチャ/10000分=約 160 時間=約 7 日)  
1アーキテクチャに対し、6G 命令を実行した際の消費電力と処理性能を評価した。

当初の予定では1M 命令に1秒ぐらい掛かると予想したが、高速 CPU で、メモリも多く搭載した LINUX マシンを使って、シングルJOB実行したため、14.4M命令/秒という実行が出来た。  
また多数のシミュレーション結果より、任意の条件にあったシステム構成を抽出することも出来た。

以下に 100 アーキテクチャを実行した結果を示す。  
各アーキテクチャの違いは、次の通りである。

- ・ 動作周波数(10 通り)
- ・ LPDDR スピード(5 通り)
- ・ LPDDR バス幅(2 通り)

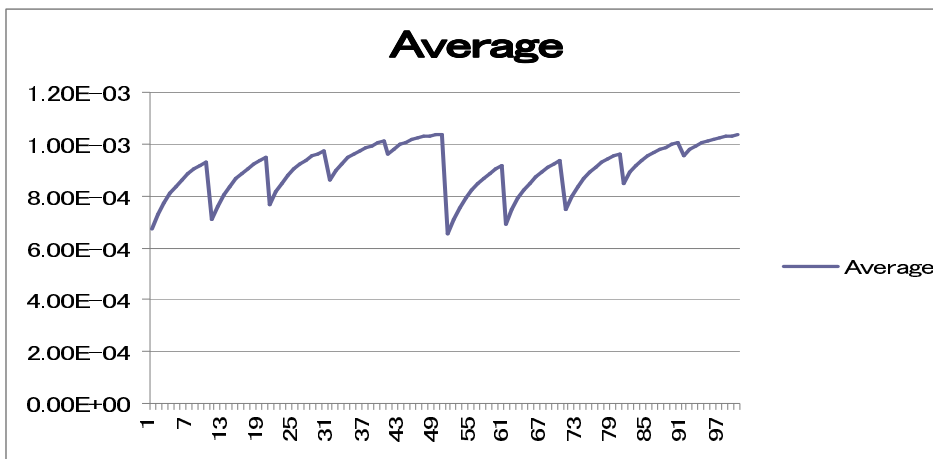


図 2-6-1 平均消費電力

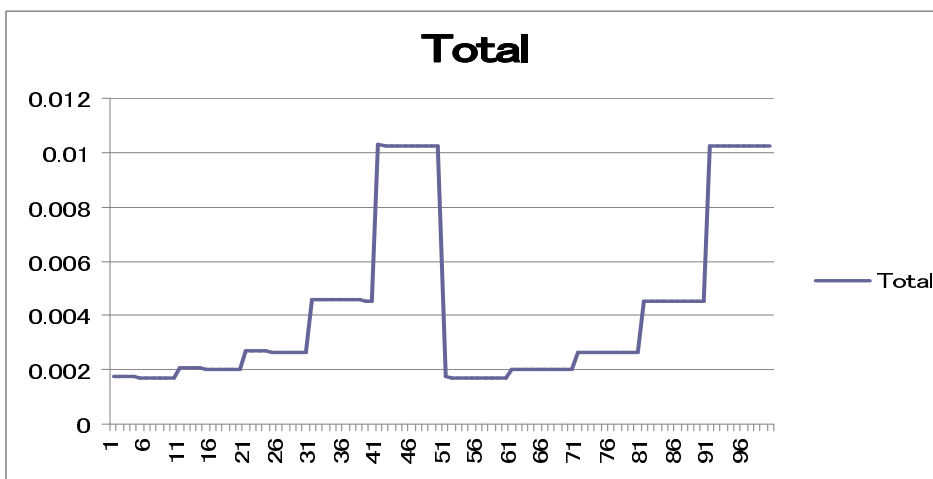


図 2-6-2 トータル消費電力

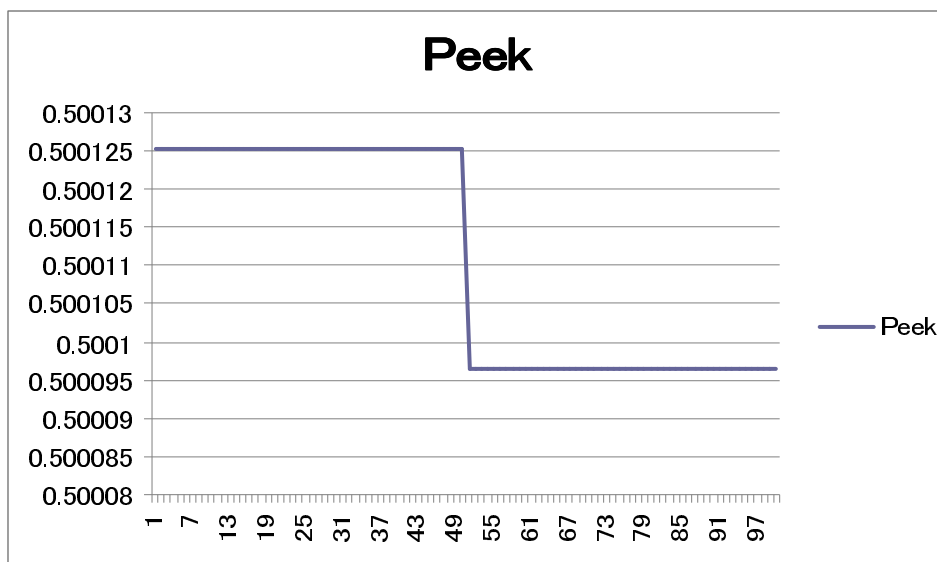


図2-6-3 最大消費電力

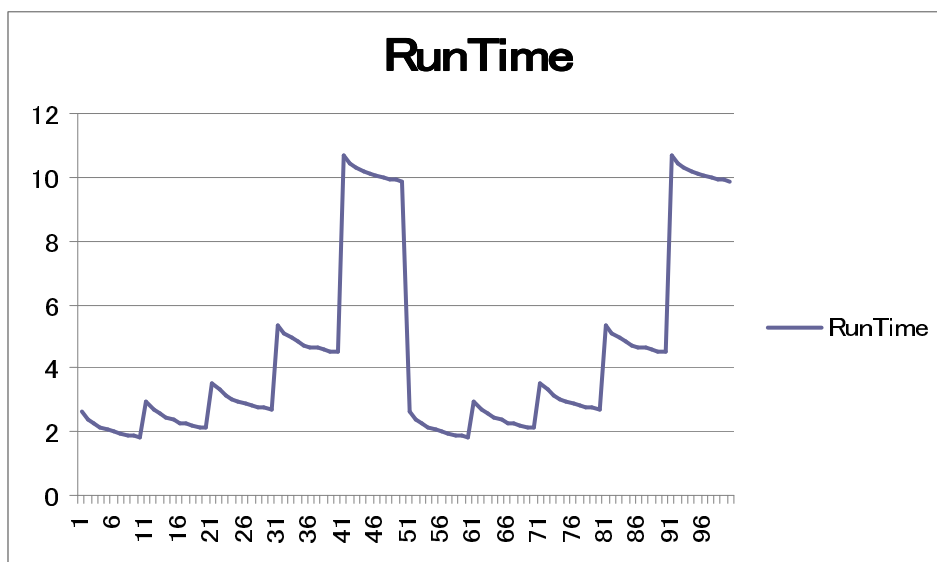


図2-6-4 シミュレーション時間

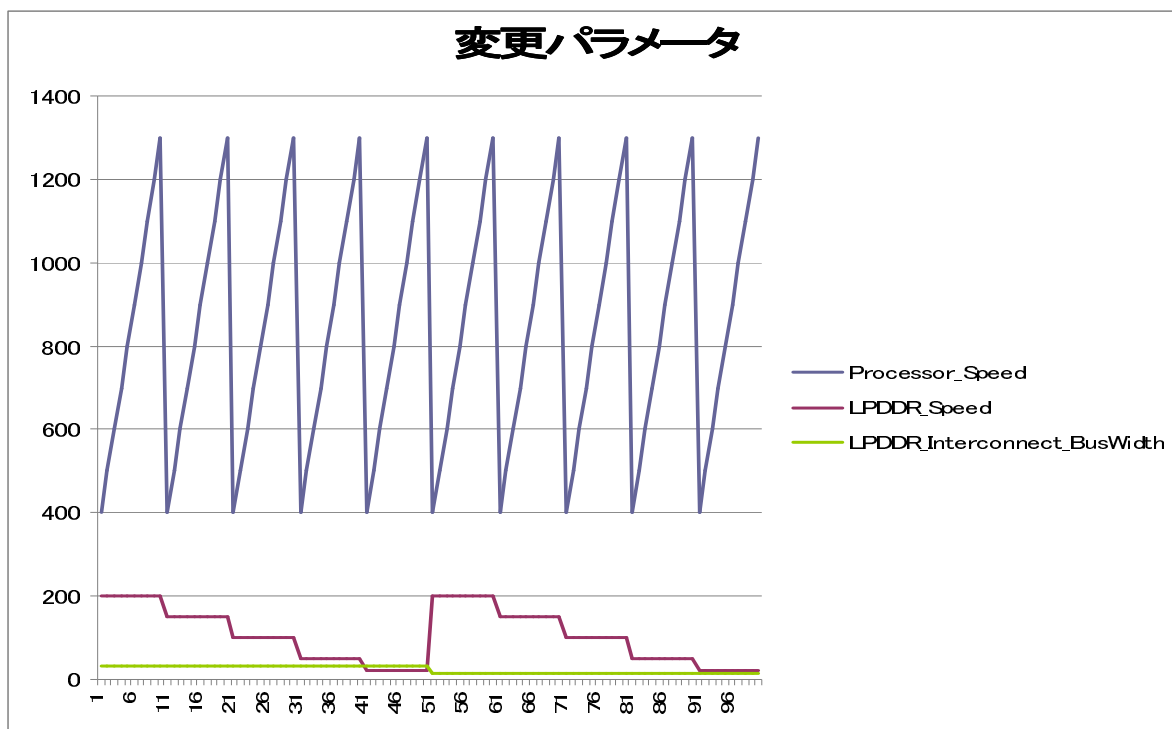


図 2-6-5 変更パラメータ値

## 2-7 実測 TEG ボードの作成・測定

### (1) 実測 TEG ボードの作製と電力測定の目的

伝送路の消費電力について、シミュレーション結果と実測値とを比較し、伝送路モデルのキャリブレーションを行うことを目的とする。

伝送路の消費電力を測定可能とするために電力測定用の端子を搭載した実測 TEG ボードを作製した。

この実測 TEG ボードには 3 種類の CPU とメモリ間の接続方法を実装し、各電力測定結果とシミュレーション結果とを比較した。

作成した「実測 TEG ボード」の概要を図 2-7-1 に示す。

(実測 TEG ボードは、実施計画書の積算表においては検査用カスタムボードという名称で記載した。)

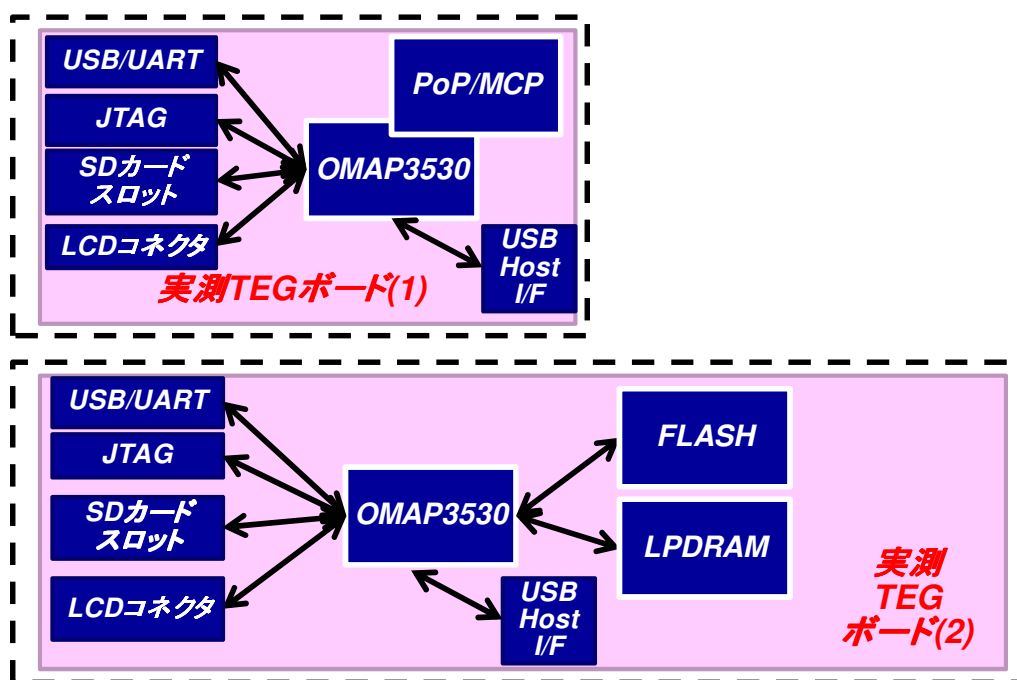


図 2-7-1 「実測 TEG ボード」概要

## (2) 実測 TEG ボードの消費電力測定項目

実測 TEG ボードを作製するにあたり、表 2-7-1 に示すように、測定項目を決定し、これに基づいて実測 TEG ボードの使用を策定した。アプリケーションの動作時の各電子部品(OMAP3530, DDR メモリ、Flash メモリ)について、コア電力(SoC 内部回路の消費電力)と I/O 電力とを分離して測定することとした。

表 2-7-1 「実測 TEG ボード」測定作業における測定項目

| ボード番号→   | 1 | 2 | #  | 測定点                               | 注意点・測定回路方式           |
|----------|---|---|----|-----------------------------------|----------------------|
| ボード全体    | ○ | ○ | 1  | 全体電力                              | ・ハイサイド電流測定回路、テストピン実装 |
| OMAP3530 | ○ | ○ | 2  | コア電力(vdd_core)                    | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 3  | コア電力(vdd_mpu_iva)                 | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 4  | コア電力(vdds_wkup_bg)                | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 5  | PLL電力(vdds_dpll_dll)              | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 6  | PLL電力(vdds_dpll_per)              | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 7  | I/O電力(vdda_dac)                   | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 8  | I/O電力(vdds)                       | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 9  | I/O電力(vdds_mmc1)<br>SDカード用        | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 10 | I/O電力(vdds_mmc1a)                 | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 11 | I/O電力(vdds_mem)<br>FlashおよびSDRAM用 | ・ハイサイド電流測定回路、テストピン実装 |
| メモリチップ   | ○ | ○ | 12 | DDR コア電力                          | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 13 | DDR I/O電力                         | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 14 | Flashコア電力                         | ・ハイサイド電流測定回路、テストピン実装 |
|          | ○ | ○ | 15 | FlashI/O電力                        | ・ハイサイド電流測定回路、テストピン実装 |

また、実測 TEG ボードを用いた消費電力測定環境の構成を図 2-7-2 に示す。

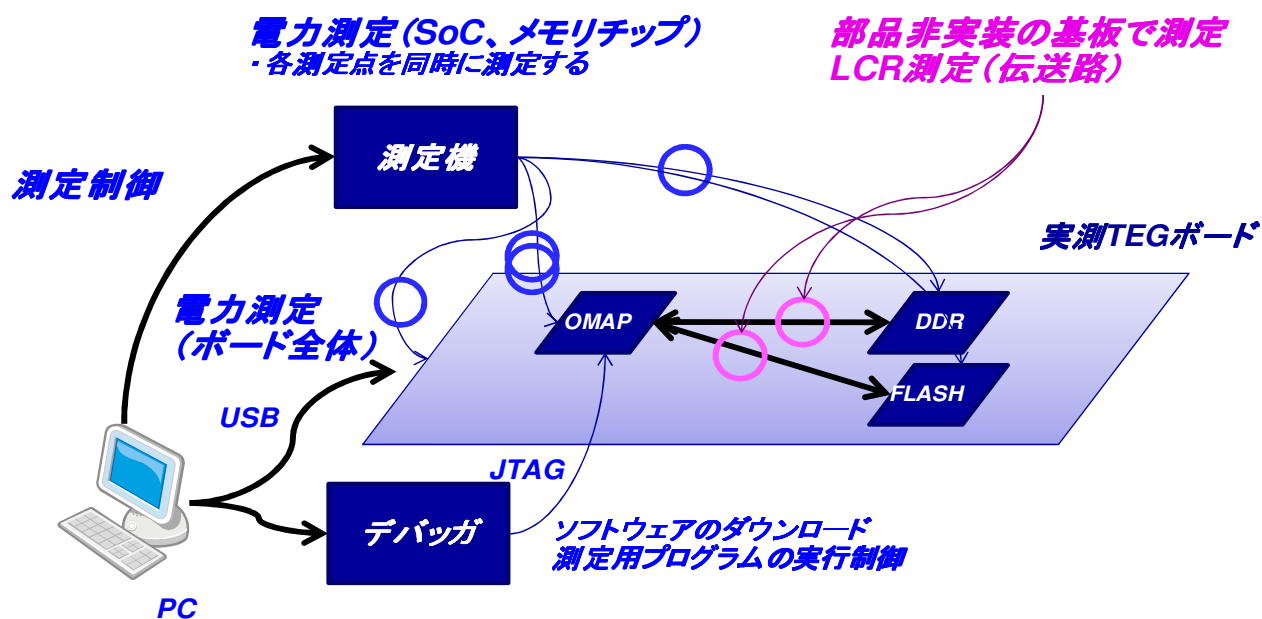


図 2-7-2 実測 TEG ボードおよび測定機器の構成

### (3) 実測 TEG ボードの消費電力測定

#### 【目的】

伝送路の消費電力を測定し、シミュレータ上の伝送路モデルによって算出する消費電力値のキャリブレーションを行うことを目的とする。

#### 【消費電力測定】

実測 TEG ボードでの伝送路の消費電力測定に先駆け、BeagleBoard を用いた全体電力の測定を行った。尚、本報告書においては BeagleBoard の全体電力の測定結果を示す。

#### <対象アプリケーション>

##### (a) 静止画の再生(デジタル・フォトフレームを想定)

使用したソフトウェア:fbv (<http://freshmeat.net/projects/fbv/>)

動作フローを図 2-7-3 に示す。

##### (b) 動画の再生

使用したソフトウェア:mplayer (<http://www.mplayerhq.hu/design7/news.html>)

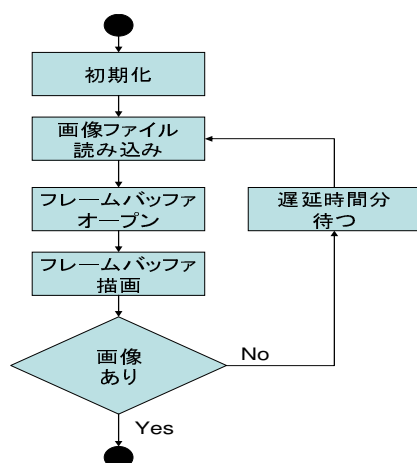


図 2-7-3 静止画表示のフローチャート

#### <測定対象>

TI 製 SoC OMAP3530 を搭載した評価ボード BeagleBoard 全体の消費電力

#### <測定方法>

電力測定回路を図 2-7-4 に示す。BeagleBoard の電源ラインにシャント抵抗(1Ω)を挿入し、電圧降下を測定することで BeagleBoard 全体の電流を測定した。電力の時間変化は、オシロスコープのログから BeagleBoard に供給されている電流と電圧を算出することで求めた。

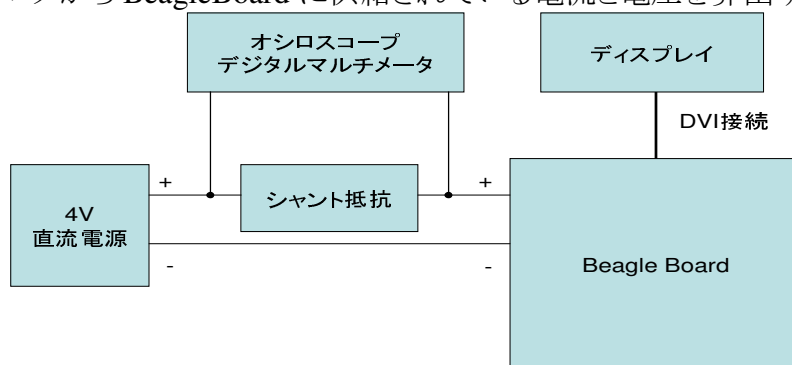


図 2-7-4 電力測定回路

- デジタルマルチメータ Instek GDM-8255A
  - 100 サンプル移動平均フィルタ 5 1/2 桁精度
- オシロスコープ Instek GDS-2202
  - サンプルング間隔 4ms サンプルング点数 12500 ポイント 5 秒間記録
  - 測定精度 2mV
- シャント抵抗 1.004  $\Omega$  (4 端子法で測定)

### 【消費電力測定結果】

BeagleBoard の消費電力の測定を行い、アプリケーションの動作に対応してシステム全体の消費電力が変化することを確認した。以下に測定結果を示す。

#### (a) 静止画の再生

静止画の再生間隔(遅延時間)を変化させた時の消費電力波形を図 2-7-5 に示す。

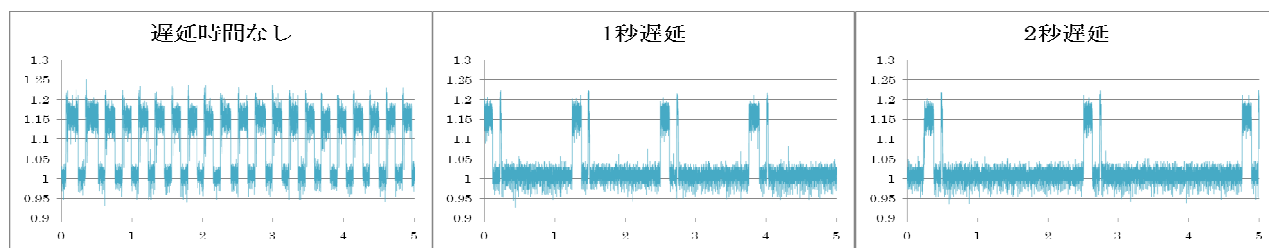


図2-7-5 静止画再生の消費電力

再生間隔(1 秒)の場合について、1 周期分の消費電力を拡大したものを図 2-7-6 に示す。

静止画再生処理のステップに応じて消費電力が変化することが分かる。処理のステップと消費電力測定結果との対応付けは、静止画再生アプリケーション・ソフトウェアのソースコード読解による。各静止画の再生処理は、「画像読み込み」、「フレームバッファオープン」、「描画」、「遅延時間分の待ち」、の流れで進行し、それぞれ図示した部分に対応すると推測した。

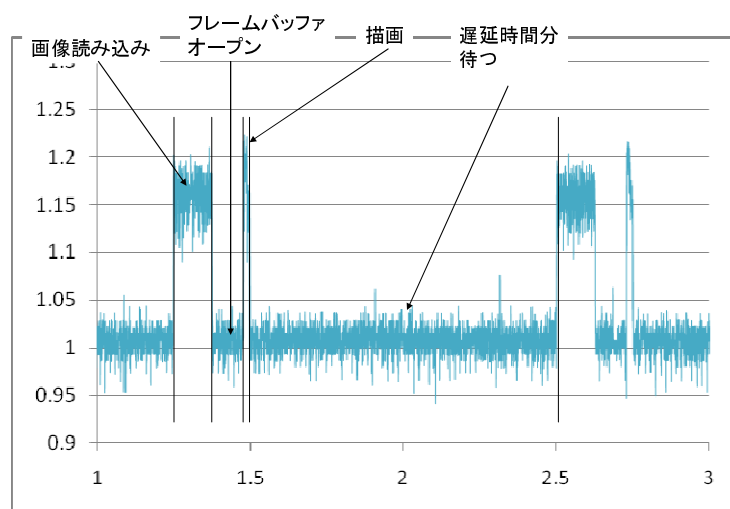


図2-7-6 静止画再生時の1 周期分の消費電力拡大(再生間隔=1 秒)

## (b) 動画の再生

動画再生時の消費電力を測定した結果を図 2-7-7 に示す。動画再生のフレームレートは 30fps、15fps、1fps の 3 種類を用いた。

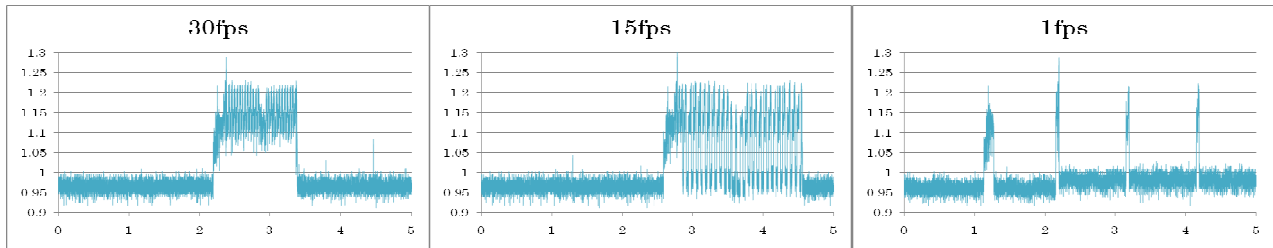


図2-7-7 動画再生時の消費電力

動画のフレームレートの違いにより、消費電力波形の変化を観測することが出来た。フレームレートを下げると、フレームの再生処理間のアイドル時間(消費電力が 0.97W 程度の期間)が長くなることが分かる。消費電力のピークは 1.3W 程度であり、ある1フレームの再生のためのデコード処理時と考えられる。消費電力の波形からフレームによって消費する電力が 1.15W~1.3W 程度まで、約 150mW のばらつきがあることが分かる。

次に、15fps の 3 周期分を拡大した結果を図 2-7-8 に示す。

動画再生の処理のステップに応じて消費電力が変化することが分かる。動画再生アプリケーション・ソフトウェアのソースコードの読解により、「フレームデコード」、「フレーム描画」、「次フレームまで待機」の3ステップに対応する消費電力であると推測した。

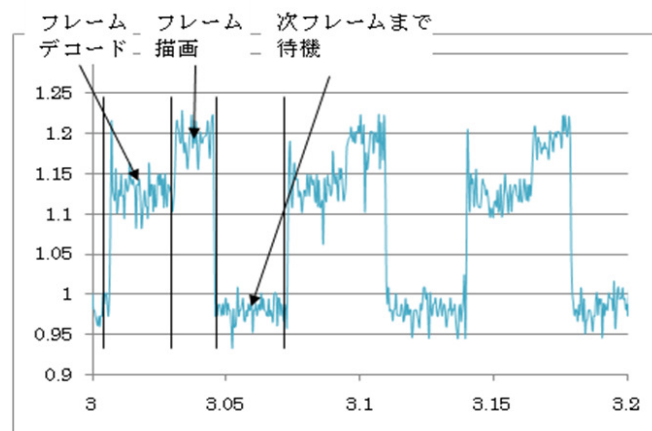


図2-7-8 動画再生時の 3 周期分の消費電力の拡大

## <消費電力測定結果まとめ>

静止画表示アプリケーションおよび動画再生アプリケーションにおける各状態の消費電力と処理時間の平均を表 2-7-2 に示す。

表 2-7-2 各状態の消費電力と処理時間の平均

| ステート  |               | 消費電力(mW) | 処理時間(ms) |
|-------|---------------|----------|----------|
| アイドル時 |               | 983      |          |
| 静止画   | 画像読み込み時       | 1150     | 128.4    |
|       | フレームバッファオープン時 | 1010     | 100.8    |
|       | 描画時           | 1180     | 21.6     |
|       | 遅延時間待ち時       | 1001     |          |
| 動画    | デコード時         | 1125     | 24.4     |
|       | 描画時           | 1188     | 14.4     |
|       | 次フレームまでの待機時   | 1018     | 27.6     |

これらの測定結果から、次のことが分かった。

- ー アイドル時の電力測定値と、アプリケーションの実行時の電力測定値には有意差(最大 200mW 程度)がある。処理時間として数 10ms から 100ms 程度の時間のスケールで消費電力状態が変化することを確認した。
- ー アプリケーションの処理ステップに応じて電力が変化する。これは、処理ステップに応じて、アクティブになる部品や部品間の通信に使用される伝送路の状態が変化する為と考えられる。
- ー アイドル時と、各アプリケーションの待機時の電力はほぼ同等である(983mW～1018mW)。これは、アプリケーション動作中でも、アイドル状態が生じることを示している。

#### (4) 車載 LAN 開発ボードの消費電力測定

##### 【目的】

本研究開発成果として製品化を予定している「Green Architect」を用いて、車載 LAN システム設計時の性能と電力のトレードオフを行えるようにすることを目的とする。

車載 LAN の通信路の消費電力と通信性能が車載 LAN システムの設計に与えるインパクトを分析するために、一般的な車載 LAN を対象とした消費電力測定を行った。

尚、車載 LAN の通信路の消費電力の伝送路としてのモデル化は、本研究開発の平成 22 年度以降に実施する予定である。

##### 【消費電力測定対象】

- ・CAN, LIN, FlexRay の 3 種類の車載 LAN の通信路

##### 【消費電力測定環境】

都築電産製の車載 LAN キット「bits pot」を使用した。この評価キットは CAN, LIN, FlexRay の 3 種類の車載 LAN に対応しており、また、サンプルアプリケーション・ソフトウェア(各種車載 LAN 通信を行ってボード間で情報を通信する、もしくは動作コマンドを送る動作を行う)が付属しており、車載 LAN ボード上に搭載されている富士通マイクロエレクトロニクス製マイコン上でサンプルアプリケーションを動作させた時の消費電力評価が可能である。

使用した車載 LAN 評価ボードの型番と、搭載されている車載 LAN 機能、マイコン名を以下に示す。

- bits pot red CAN-MOTOR CAN-100  
CAN・モータ制御 FR-60Lite
- bits pot yellow LIN LIN-100  
LIN 通信と F2MC-8FX シリーズ
- bits pot white CAN-LIN CAL-10  
CAN-LIN 通信と F2MC-16FX シリーズ
- bits pot blue FlexRay FLR-100  
FlexRay と FR60 MB91460 シリーズ

### 【車載 LAN の通信路の消費電力測定結果】

車載 LAN ボードでボード間の通信を行うサンプル・アプリケーションを動作させ、モータの回転制御や車載 LAN 通信などの条件の組合せに対して消費電力を測定した結果を表 2-7-3 に示す。また、測定結果の妥当性を評価するためにデータシートの仕様と比較した結果を表 2-7-5 と表 2-7-6 に示す。

表 2-7-3 車載 LAN ボードの消費電力測定結果

| 基板     | 動作モード              | その他     | 消費電力(mW) | シャント抵抗<br>電圧(mV) | シャント抵抗<br>電流(mA) |
|--------|--------------------|---------|----------|------------------|------------------|
| Red    | リセット時              |         | 124      | 28               | 25               |
|        | 単体動作               | モータ回転   | 692      | 156              | 138              |
|        |                    | モータ停止   | 636      | 143              | 127              |
|        |                    | モータブレーキ | 664      | 150              | 133              |
|        | CAN受信(相手なし)        |         | 600      | 135              | 120              |
|        | CAN受信(相手あり)        | モータ回転   | 711      | 160              | 142              |
|        |                    | モータ停止   | 608      | 137              | 122              |
|        |                    | モータブレーキ | 648      | 146              | 130              |
| Blue   | リセット時              |         | 151      | 34               | 30               |
|        | 単体動作               |         | 514      | 116              | 103              |
|        | FlexRayNode1(相手なし) |         | 552      | 125              | 110              |
|        | FlexRayNode1(相手あり) |         | 571      | 129              | 114              |
|        | FlexRayNode2(相手なし) |         | 561      | 127              | 112              |
|        | FlexRayNode2(相手あり) |         | 568      | 128              | 114              |
| Yellow | リセット時              |         | 170      | 38               | 34               |
|        | 単体動作               |         | 229      | 52               | 46               |
|        | LINスレープ(相手なし)      |         | 222      | 50               | 44               |
|        | LINスレープ(相手あり)      | 通信なし    |          |                  |                  |
|        |                    | 通信あり    | 208      | 47               | 42               |
| White  | リセット時              |         | 30       | 7                | 6                |
|        | 単体動作               |         | 244      | 55               | 49               |
|        | LINマスタ(相手なし)       |         | 141      | 32               | 28               |
|        | LINマスタ(相手あり)       | 通信なし    | 0        |                  | 0                |
|        |                    | 通信あり    | 168      | 38               | 34               |
|        | CAN送信(相手なし)        |         | 209      | 47               | 42               |
|        | CAN送信(相手あり)        | 通信なし    | 145      | 33               | 29               |
|        |                    | 通信あり    | 155      | 35               | 31               |

車載 LAN 通信時の通信速度と消費電力から、通信量あたりのエネルギー効率を算出した結果を表 2-7-4 に示す。この結果から、通信量あたりのエネルギー効率は LIN の場合が最も悪く、FlexRay の場合が最も良いということが分かる。なお、FlexRay1 と FlexRay2 は両方とも送信・受信を行うプログラムであるため、2 つのボードのデータを示した。

ただし、この結果は車載 LAN 通信路のみの消費電力ではなく、車載 LAN 制御用のマイコンの消費電力も含まれているため、マイコンの動作周波数や半導体テクノロジーの違いなどによると考えられる。

表 2-7-4 通信量あたりのエネルギー効率比較

| 伝送路      | マイコン<br>動作周波数<br>(MHz) | 消費電力(mW) |       |                 | 通信量(Byte/s) | エネルギー効率<br>( $\mu$ J/Byte) |
|----------|------------------------|----------|-------|-----------------|-------------|----------------------------|
|          |                        | 通信なし     | 通信あり  | 差分<br>(通信による増分) |             |                            |
| CAN      | 33                     | 691.7    | 711.1 | 19.5            | 8000        | 2.4                        |
| LIN      | 20                     | 140.6    | 167.6 | 27.0            | 202         | 133.7                      |
| FlexRay1 | 100                    | 552.5    | 570.7 | 18.2            | 32000       | 0.6                        |
| FlexRay2 | 100                    | 561.4    | 567.6 | 6.1             | 32000       | 0.2                        |

表2-7-5 車載LAN ボードの消費電力測定結果と仕様上の消費電力値の比較(1)

|                   |                            | 最大電力   | ランモード                                              | スリープモード                                                  | 実測値                             |
|-------------------|----------------------------|--------|----------------------------------------------------|----------------------------------------------------------|---------------------------------|
| RED<br>(CAN)      | FRシリーズ<br>MB91F267N        | 600mW  | 33MHz<br>標準90mA<br>最大100mA<br>Vcc=4.0~5.5V         | 33MHz<br>標準60mA<br>最大80mA<br>Vcc=4.0~5.5V                | リセット時<br>124mW<br>(25mA)        |
|                   | CANTランシーバIC                |        |                                                    |                                                          | 動作時<br>600~711mW<br>(120~142mA) |
| BLUE<br>(FlexRay) | FRシリーズ<br>MB91F465X        | 1000mW | ICC (100MHz)<br>標準125mA<br>最大155mA<br>Vcc=4.5~5.5V | ICCH(STOP)<br>標準30uA<br>最大150uA<br>Vcc=4.5~5.5V<br>@25°C | リセット時<br>151mW<br>(30mA)        |
|                   | 電源監視用Reset IC<br>MB3793-42 |        | ICC1<br>標準27uA<br>最大50uA<br>Vcc=5.0V               |                                                          | 動作時<br>514~571mW<br>(103~114mA) |

表2-7-6 車載LAN ボードの消費電力測定結果と仕様上の消費電力値の比較(2)

|                        |                                                          | 最大電力                                                                                       | ランモード                                                                                                                            | スリープモード                                                                                                                              | タイマーモード                                                                                                                            | 実測値                                                         |
|------------------------|----------------------------------------------------------|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| YELLOW<br>(LIN)        | F <sup>2</sup> MC-8FXシリーズ<br>MB95F136JBS<br>(UART,LIN搭載) | 320mW                                                                                      | ICC (20MHz)<br>標準9.5mA<br>最大12.5mA<br>ICC (32MHz)<br>標準15.2mA<br>最大20.0mA<br>Vcc=4.5~5.5V                                        | ICC (20MHz)<br>標準4.5mA<br>最大7.5mA<br>ICC (32MHz)<br>標準7.2mA<br>最大12.0mA<br>Vcc=4.5~5.5V                                              |                                                                                                                                    | リセット時<br>170mW<br>(34mA)                                    |
|                        | F2MC-8LXシリーズ<br>MB89P585B<br>(USB搭載)                     | 300mW                                                                                      | 12MHz<br>標準25mA<br>最大38mA<br>Vcc=5.0V                                                                                            | 12MHz<br>標準20mA<br>最大30mA<br>Vcc=5.0V                                                                                                |                                                                                                                                    | 動作時<br>208~228mW<br>(52~45mA)                               |
| WHITE<br>(CAN-<br>LIN) | F2MC-16FXファミリ<br>MB96F356<br>0.18um CMOS                 | 許容消費電力<br>(フラッシュデバ<br>イス)<br>400mW<br>(125°C)<br>560mW(115°C)<br><br>Vcc=3.0~5.5V<br>@25°C | ・ICCPPLL<br>(96MHz)<br>標準49mA<br>最大62mA<br>・ICCMMAIN<br>(4MHz)<br>標準4.5mA<br>最大5.5mA<br>・ICCSUB<br>(32KHz)<br>標準0.1mA<br>最大0.2mA | ・ICCPPLL<br>(96MHz)<br>標準15mA<br>最大16.5mA<br>・ICCMMAIN<br>(4MHz)<br>標準1.5mA<br>最大1.8mA<br>・ICCSUB<br>(32KHz)<br>標準0.04mA<br>最大0.12mA | ・ICCPPLL<br>(48MHz)<br>標準1.6mA<br>最大2mA<br>・ICCMMAIN<br>(4MHz)<br>標準0.35mA<br>最大0.5mA<br>・ICCSUB<br>(32KHz)<br>標準0.03mA<br>最大0.1mA | リセット時<br>30mW<br>(6mA)<br><br>動作時<br>141~244mW<br>(28~49mA) |

#### ＜車載 LAN の通信路の消費電力測定方法＞

車載 LAN ボードの消費電力測定システムの構成を図 2-7-9に示す。

車載 LAN ボード全体の消費電力を測定するために、USB 給電の電源ラインにシャント抵抗を挿入し、マルチメータおよびオシロスコープを用いて降下電圧から電流を算出した。シャント抵抗は 1.127Ω(4 端子法による測定結果)のものを用いた。

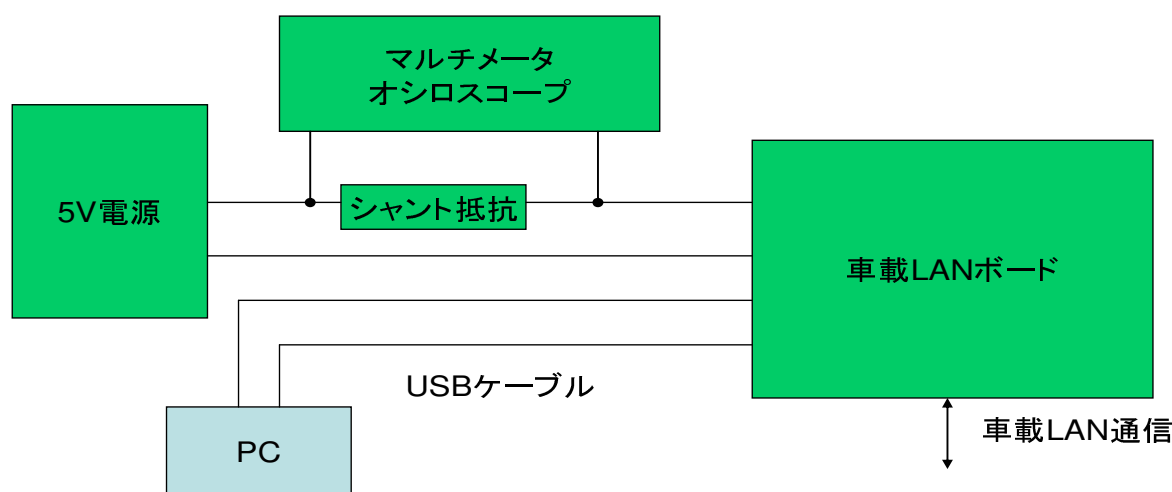


図2-7-9 車載LAN ボード消費電力測定システム構成図

### 第3章 全体総括

#### 3-1 研究開発成果

本研究開発により、以下の成果を得ることができた。

##### 【本研究開発の最終目標とその達成度】

- (1) システムレベルの消費電力シミュレーション性能として、以下の目標を達成した。  
ーシミュレーション速度:1M命令/秒  
平成21年度の目標を達成した。(シミュレーション速度:7M命令/秒以上)
- (2) 最適なシステム構成の探索を可能とする。  
平成21年度の達成度:最適なシステム構成の探索を可能とするためのシミュレーション環境の構築を完了した。

情報機器のシステムレベルの消費電力をシミュレーションするために、以下の開発を完了した。

- ・シミュレーション用の各種モデル開発(電子部品、伝送路)を完了した。
- ・ターゲットシステムのハードウェア及びソフトウェアのシミュレーション環境の構築を完了した。
- ・システムレベルでの消費電力シミュレーション設計環境の構築を完了した。

尚、平成22年度以降も研究開発を継続し、アーキテクチャ探索により、消費電力の最適化設計のフロントローディング化を実現し、製品設計における手戻りを50%削減し、トータルの開発期間を30%短縮することを目標とする。

平成21年度の技術的目標値を以下のように設定し、目標を達成した。

表3-1-2 平成21年度の技術的目標

| 平成21年度の概要                             | 技術的目標値                                                                                                          |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 平成21年度<br>「基本システムの完成と高速シミュレーション手法の確立」 | 10種類の伝送路モデル、3つのアーキテクチャモデルを作成し、3つのアプリケーションで電力評価を行った。<br>アーキテクチャ探索システムを実装し、機能と性能のテストを行った。シミュレーション速度1M命令/秒以上を達成した。 |

また、各サブテーマの技術的目標値は以下のように達成した。

表3-1-3 平成21 サブテーマの技術的目標と達成度

| テーマ                          | 技術的目標値達成                                                                                            |
|------------------------------|-----------------------------------------------------------------------------------------------------|
| ※ 伝送路モデルの開発                  |                                                                                                     |
| 1.1. 伝送路モデルの開発               | 10種類の伝送路モデルを開発した。                                                                                   |
| 1.2. システムレベルシミュレータへの統合       | 伝送路を含めたシステムレベルシミュレーションを可能にした。                                                                       |
| 1.3. アプリケーションソフト動作時のシステム電力評価 | 3つのアーキテクチャを想定してモデルを作成し、3つのアプリケーションで電力を評価した                                                          |
| ※ アーキテクチャ探索システムの開発           |                                                                                                     |
| 2-1. 実装組合せ制約条件に基づく網羅方法の検討    | 具体的なシステム(例:スマートフォン)を例にして、データベースの部品の全組み合わせから、制約条件を基にした網羅アルゴリズムをソフトウェアとして実装し動作させた。                    |
| 2-2. 探索システムの開発               | 部品データベース・伝送路データベース・アプリケーションデータベースのデータを基にして、消費電力・処理性能シミュレーションを行い、結果を解析結果データベースに格納する、探索システムを実装し動作させた。 |

|             |                                                                                 |
|-------------|---------------------------------------------------------------------------------|
| 2-3. テスト・検証 | 探索システムの機能と性能のテストを行った。1アーキテクチャに対し、6G 命令を実行した際の消費電力と処理性能を、6000 秒以内に評価完了することを確認した。 |
|-------------|---------------------------------------------------------------------------------|

### 3-2 課題

平成 21 年の研究開発の結果、次の課題と対策案が明確となった。これらの課題は、平成 22 年度以降の研究開発の中で対策案を遂行することにより、より確度の高い研究開発を進める。

表 3-2-1 平成 21 年度の研究開発の課題と対策案

| 分類                             | 課題                                                                            | 対策案                                                                          |
|--------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 1 伝送路モデルの開発                    |                                                                               |                                                                              |
| 1-1 伝送路モデルの開発                  | CAN、LIN、FlexRay 等のシステム・レベルの伝送路のモデルの開発は仮モデルで対応した。<br>(原因:当初からモデルの作成は予定していなかった) | 平成 22 年度以降の研究開発の中でモデル化と実測値との比較によるモデルのキャリブレーションを進める。                          |
| 1-2 システムレベルシミュレータへの統合          | 特になし                                                                          | 特になし                                                                         |
| 1-3 アプリケーションソフトウェア動作時のシステム電力評価 | 計画時から、測定時のアプリケーションが3種類であり、測定は行うことが出来た。                                        | より多くのアプリケーションにて電力評価を行う。                                                      |
| 2 アーキテクチャ探索システムの開発             |                                                                               |                                                                              |
| 2-1 実装組合せ制約件に基づく網羅方法の検討        | 特になし                                                                          | 特になし                                                                         |
| 2-2 探索システムの開発                  | 制約情報の項目を2つ設定した。<br>Linux マシンで実行出来る環境を作成した。                                    | 制約情報の追加に柔軟に対応出来るように機能追加する。<br>Windows 環境で実行できものを開発する。(Linux->Windows ポータリング) |
| 2-3 テスト・検証                     | 特になし                                                                          | 特になし                                                                         |

### 3-3 事業化展開

本研究開発の完了は平成 23 年度であり、当初の予定通り平成 24 年度よりアーキテクチャ探索システム(製品名「Green Architect」)の国内での販売を開始する。

平成 24 年度以降の売り上げ目標を以下のグラフに示す。

売り上げ目標:10 億円@2016 年

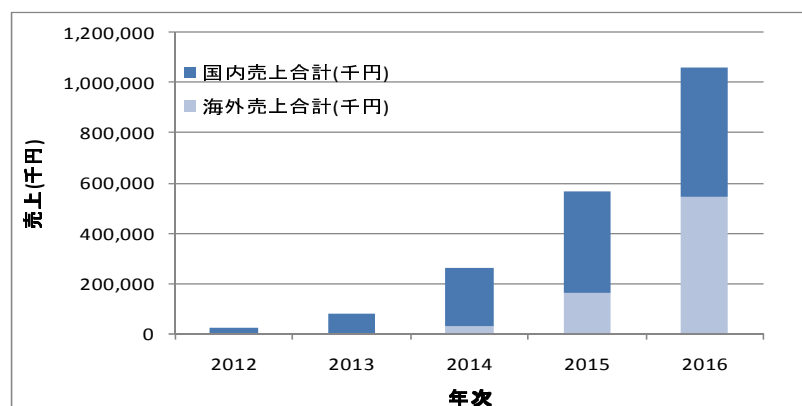


図 3-3-1 売り上げ目標

売り上げは以下の3項目について計上した。

- ・ライセンス販売:1ユーザライセンスを 500 万円として販売する
- ・ライセンス保守:ライセンス販売後1年ごとに最新版へのアップデートに対して課金する。  
(単価 75 万円)
- ・カスタムモデリングサービス:ユーザの要求に応じた伝送路モデルの設計サービス。  
(単価 250 万円)

表3-3-1 売り上げの内容

|                       |                 |        |             | 2009-2010 | 2011       | 2012            | 2013    | 2014    | 2015    | 2016      |         |
|-----------------------|-----------------|--------|-------------|-----------|------------|-----------------|---------|---------|---------|-----------|---------|
|                       |                 | 分担     | 展開フェーズ      | 研究開発・製品化  | ユーザテスト(B版) | 国内販売開始          |         | 海外販売開始  |         |           |         |
| 国内                    | ライセンス販売         | Keirex | 単価(千円)      | 開発中       |            | 5,000           | 5,000   | 5,000   | 5,000   | 5,000     |         |
|                       |                 |        | 国内販売本数(本)   |           |            | 3               | 10      | 30      | 50      | 60        |         |
|                       | 売上(千円)          |        | 15,000      |           |            | 50,000          | 150,000 | 250,000 | 300,000 |           |         |
|                       | 単価(千円)          |        | 750         |           |            | 750             | 750     | 750     | 750     |           |         |
|                       | ライセンス保守         | Keirex | 国内累積販売本数(本) |           |            | 0               | 3       | 13      | 43      | 93        |         |
|                       |                 |        | 売上(千円)      |           |            | 0               | 2,250   | 9,750   | 32,250  | 69,750    |         |
|                       | カスタムモデリングサービス   |        | TOPS        |           |            | 単価(千円)          | 5,000   | 5,000   | 5,000   | 5,000     | 5,000   |
|                       |                 |        |             |           |            | 件数(件)           | 2       | 5       | 15      | 25        | 30      |
|                       |                 |        |             |           |            | 売上(千円)          | 10,000  | 25,000  | 75,000  | 125,000   | 150,000 |
|                       |                 |        |             |           |            | 国内売上合計(千円)      | 25,000  | 77,250  | 234,750 | 407,250   | 519,750 |
| 海外<br>(海外代理店経由での取引予定) | ライセンス販売         |        | Keirex      |           |            | 単価(千円)※販売手数料差引後 |         |         | 3,500   | 3,500     | 3,500   |
|                       |                 |        |             |           |            | 国内販売本数(本)       |         |         | 5       | 30        | 100     |
|                       | 売上(千円)          |        |             |           |            |                 | 17,500  | 105,000 | 350,000 |           |         |
|                       | 単価(千円)※販売手数料差引後 |        |             |           |            |                 | 525     | 525     | 525     |           |         |
|                       | ライセンス保守         | Keirex | 国内累積販売本数(本) |           |            |                 |         | 0       | 5       | 35        |         |
|                       |                 |        | 売上(千円)      |           |            |                 |         | 0       | 2,625   | 18,375    |         |
|                       | カスタムモデリングサービス   |        | TOPS        |           |            | 単価(千円)※販売手数料差引後 |         |         | 3,500   | 3,500     | 3,500   |
|                       |                 |        |             |           |            | 件数(件)           |         |         | 3       | 15        | 50      |
|                       |                 |        |             |           |            | 売上(千円)          |         |         | 10,500  | 52,500    | 175,000 |
|                       |                 |        |             |           |            | 海外売上合計(千円)      |         |         | 28,000  | 160,125   | 543,375 |
| 合計売上(千円)              |                 |        |             |           |            | 25,000          | 77,250  | 262,750 | 567,375 | 1,063,125 |         |

また、海外への販売に際しては、現地の販売代理店を経由して販売・サポートを行う予定である。

以下に、売上見積もりの根拠を示す。

(出典:「NEC、携帯基幹部品を共通化 開発費、1機種当たり4割減」 日経 IT+PLUS [2009年1月27日/日本経済新聞 朝刊]

<http://it.nikkei.co.jp/mobile/news/index.aspx?n=AS1D23017%2026012009>)

その他のメーカーでも、以下の図で示すように多くの機種を開発している。

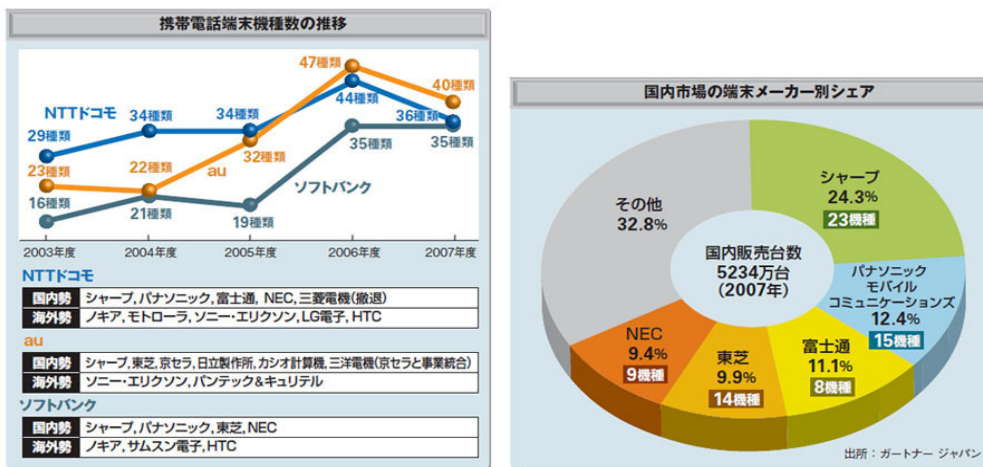


図3-3-2 携帯端末開発状況

(出典:日経 BP 社 ITPro <http://itpro.nikkeibp.co.jp/article/COLUMN/20080717/311012/>)

また、市場の情報機器の販売動向を調査すると、現在販売されている携帯電話の機種は 500 を超え、参入しているセットメーカー(機器開発メーカー)の数は 20 を超えている。その他の情報機器も多数の開発メーカーが参入している。

表 3-3-2 情報機器販売状況(出典:価格.com 2009 年 6 月 29 日現在)

|            | 販売機種数(種類) | 開発メーカー数(社) |
|------------|-----------|------------|
| 携帯電話       | 563       | 21         |
| 液晶テレビ      | 1177      | 47         |
| デジタルカメラ    | 317       | 25         |
| デジタルビデオカメラ | 188       | 15         |

一般消費者、製品提供者(製造会社、設計者)が、低消費電力設計に対する認識を十分理解し、その価値を認めることで、GreenArchitect のライセンス販売量は年間 50～60 本が見込まれる。

## 付録

### A 専門用語等の解説

#### 【命令セットシミュレータ】

ISS(Instruction Set Simulator)。マイクロプロセッサ(CPU)の振る舞いを模倣するシミュレーション モデルの一種であり、ユーザ・インターフェイスを加えてシミュレータとして構成されることが多い。ISS は、通常高級言語でプログラムされており、シミュレートするマイクロプロセッサの命令セット・アーキテクチャの機械語命令を読み込み、プロセッサのレジスタを表現する内部変数の状態を維持して命令レベルでの動作を模倣する。

#### 【機械語】

マイクロプロセッサ(CPU)が直接理解し実行できるプログラミング言語である。機械語は 2 値の電気信号の集まりとして定義され、その解釈は電気回路として実装される。人が使う場合は 16 進数などで表現される、抽象度の最も低いレベルのプログラム言語といえる。

#### 【QEmu】

QEMU は、数多くのマイクロプロセッサ(CPU)の動作を模倣するシミュレータであり、汎用性がありオープンソースであるという特徴がある。

### B 引用文献

- ・日経 IT+PLUS [2009 年 1 月 27 日/日本経済新聞 朝刊]  
<http://it.nikkei.co.jp/mobile/news/index.aspx?n=AS1D23017%2026012009>
- ・日経 BP 社 ITPro <http://itpro.nikkeibp.co.jp/article/COLUMN/20080717/311012/>