Vol.2014-SLDM-165 No.29
Vol.2014-EMB-32 No.29
2014/3/16

書換処理スケジューラによる電子ペーパ搭載端末の省電力化手法

城田祐介[†1]　吉村礎[†1]　瀬川淳一[†1]　金井達徳[†1]

昨今の省電力技術への期待の高まりの背景の下，電源を遮断しても表示を保持できるディスプレイである電子ペーパが注目されている．電子ペーパを利用するとシステムのアイドル時の消費電力を大幅に下げられるため，超低消費電力な端末を実現できる可能性がある．一方で，電子ペーパのような不揮発性ディスプレイでは，電源を遮断しても表示を保持できる安定した状態間を切り替える必要があり，書換え時に大きな電力を要する．そのため，超低消費電力化を目指す上では，アクティブ時の消費電力の削減も重要な課題である．本稿では，電子ペーパ搭載端末のアクティブ時の消費電力を削減するために，電子ペーパに不向きな書換え命令列を，小さい電力で書換え可能な命令列に再構成する書換え制御方式を提案する．デバイスドライバレベルで動的に再構成することでアプリケーションの変更なく低消費電力化を実現できる．

Exploiting Electronic Paper Display Update Scheduler in Extremely Low Power Mobile Platforms

YUSUKE SHIROTA[†1]　SHIYO YOSHIMURA[†1]
JUN'ICHI SEGAWA[†1]　TATSUNORI KANAI[†1]

EPD (Electronic Paper Display) is a widely used non-volatile display which can retain images on screen even with powers removed. While exploiting EPD can significantly reduce power consumption for idle durations, updating such non-volatile displays requires much energy switching between bistable states. Therefore, power reduction during EPD updates is an important issue. This paper introduces EPD update scheduler which dynamically rearranges update requests ill-suited for EPDs to localized and collision-free low power consuming requests in the device driver level, thus realizing extremely low-power without need for application modifications.

1. はじめに

近年，携帯型のタブレット端末やウェアラブル型のスマートウオッチなどの情報処理端末において，電力利用の効率化が重要な技術課題となっている．これらの端末においては, 全体の消費電力のうち LCD などの表示装置のバックライトやリフレッシュが占める割合が高いことが知られている[1]．それに呼応して，不揮発性メモリをフレームバッファに利用するディスプレイ[2]や, 1 Hz 程度の低いリフレッシュレートで静止画を表示可能な IGZO [3]やメモリ液晶などの低消費電力化したディスプレイが続々登場している．その中でも，電子ペーパ(EPD: Electronic Paper Display)[4]は，表示内容に変化がない場合には電源を遮断しても表示を保持できる不揮発性のディスプレイとして注目されており，様々な EPD 方式が存在する[5][6][7]．EPD 技術は進化を続けており，Kindle[8]などの読書に特化した電子書籍端末で広く普及している E Ink[5]の電気泳動方式では画面を白黒反転させるリフレッシュ処理を削減する改良がなされたり，Mirasol[6][9]などのカラーで高速な EPD も登場してきている．また，EPD と LCD を 1 つの端末に持つ*Android ™ OS を搭載したスマーフォン[10]なども製品化されている. 情報処理端末のディスプレイに不揮発性の EPD を採用することで，電子ペーパ搭載端末でドキュメントワークを行っている最中でもユーザが画面を閲覧している入力待ち状態では，LCD のように定期的にリフレッシュする必要がない．そのため，表示を続けた状態でプロセッサを低消費電力モードに移行させてアイドル時の消費電力を下げることで，不揮発性を活かして情報処理端末の消費電力を大幅に低減することができ，これまでにないような超低消費電力な端末を実現できる可能性が出てきている．

しかし，不揮発性ディスプレイの場合，書換え処理は，電源を遮断しても表示を保持できる安定した状態間を切り換えるため，大きな電力が必要となる．より小さい電力で EPD 搭載端末を駆動可能にしてバッテリ寿命の向上や小さいバッテリで軽量化を実現するためにも，単に既存の LCD を EPD で置き換えるだけでは十分でなく，EPD の書換え処理に掛かる消費電力を最小化する細やかな低消費電力制御が重要である．

本稿では，書換え命令を実行時にデバイスドライバレベルで再構成する EPD スケジューラを提案する．EPD スケジューラでは，例えば，ドキュメントビューアから，重ねて表示される文章とページ番号の 2 つの書換え命令が短い間隔で発行されると，1 回の書換え命令で同時に表示されるように再構成する．EPD では，命令間で書換え領域に重

†1 (株)東芝　研究開発センター
TOSHIBA Corporation, Corporate Research & Development Center

Vol.2014-SLDM-165 No.29
Vol.2014-EMB-32 No.29
2014/3/16

なりがある場合，書換え処理が遅延する特性がある．そのため，ユーザが気にならない程度の短い時間だけ後続の書換え命令を待ち，重なりのない小さな電力で書換え可能な命令に変換する．

以下，2 章で EPD 搭載端末における EPD 書換え処理の概要を述べ，3 章でその課題について述べる．4 章では EPD 書換え処理の低消費電力制御方式を提案し，5 章で我々の開発した EPD 搭載端末評価ボード上で性能評価を行う．6 章で今後の課題ついて述べ，7 章でまとめる．

2. 電子ペーパ書換え処理の概要

本章では，EPD 書換え処理の低消費電力制御方式の評価で用いる E Ink EPD 搭載端末評価ボード上での書換え処理シーケンスの概要を説明する．

我々の開発した評価ボードでは，EPD コントローラを内蔵した i.MX508[11]アプリケーションプロセッサを利用している．

まず，i.MX508 上で動作するアプリケーションが，EPD に表示する書換え用データを主記憶上の RGB 形式のフレームバッファに書き込む．LCD と異なり，EPD には定期的なリフレッシュがないので，アプリケーションが明示的に書換え命令を EPD コントローラのデバイスドライバに発行する．

次に，デバイスドライバが，アプリケーションから指定された書換え用データをアクセラレータである ePXP(Enhanced Pixel Pipeline)を利用して RGB から EPD で表示可能なグレースケール色空間への変換などの EPD 特有の前処理を実行する．前処理済みデータは，主記憶上に確保された EPD コントローラ専用のワーキングバッファにコピーされ，そこから書換え処理が実行される．

EPD は，現在の LCD と異なり，EPD コントローラにより画面を部分的に書換えられるのが特徴の一つである．i.MX508 プロセッサの EPD コントローラは，16 個の書換えエンジン(LUT)を有しており，矩形の部分領域を並列にかつ非同期的に書換え可能である．

3. 電子ペーパ書換え処理の課題

本章では，EPD の書換え時に消費電力が増大する要因について述べる．

3.1 断続的に発行される書換え命令による書換え処理時間の増加

タブレット端末などで動作する既存の多くのアプリケーションを EPD 搭載端末上で利用できることが望ましい．しかし，これらは LCD を前提に作成されており，LCD の高速に書換え可能な特性を利用して操作感を高めるために，準備できたものから表示して応答性を高めたり，様々なエフェクトを付加している．そのため，これら LCD 向けアプリケーションがフレームバッファに表示データを書き込む時に単純に EPD に書換え命令を発行するだけの改良では，アプリケーションから書換え命令が断続的に発行されてしまう．

EPD は，デバイスの特性上，書換え処理時間(EPD コントローラが動作中の時間)が LCD と比較すると非常に長いのが特徴である．E Ink EPD には，複数の書換えモードがあり，残像が残るが書換え時間が比較的短い高速書換えモードもあるが，残像を残さずに書換える通常書換えモードでは，我々の評価ボード上で 1 秒程度掛かる．そのため，断続的に書換え命令が到着するとその間，常に EPD が書換え処理中のままになってしまう．

図 1 で，画像ビューアからサムネイル表示の際に複数の画像の書換え命令 A，B，C が短期間に集中して発行される例を用いて説明する．EPD コントローラの既存のデバイスドライバは，書換え命令を画像ビューアから受信するとすぐに EPD コントローラに書換え処理開始を指示する．すると，命令 A の書換え処理開始から命令 C の書換え処理終了までの間，EPD だけでなく，EPD コントローラを内蔵している SoC，EPD コントローラのワーキングバッファがある主記憶，EPD に電源供給している EPD 用の PMIC などが常時アクティブになり，消費電力が大きくなってしまう．

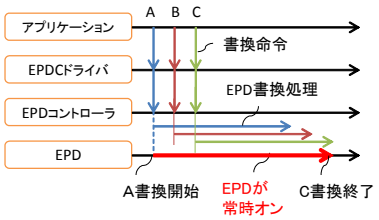

図 1　集中して発行される書換え命令による
EPD 書換え処理時間の増加

Figure 1　Update time increases with converging updates.

3.2 コリジョンによる書換え処理時間の増加

EPD の各々の書換え処理は非常に時間が掛かるが，複数の書換え命令が連続して発行された場合，EPD コントローラの複数の LUT をうまく利用して並列に書換え処理すれば動きのある表示も見せることもできる．しかし，複数命令間で表示領域が重なる場合，EPD コントローラ内部で書換え処理の衝突（コリジョン）が検出され，先行する書換え処理の完了を待ってから重なった領域の書換え処理を再実行する必要がある．そのため，書換え処理時間が更に長くなり，消費電力のさらなる増大をもたらす．

コリジョンが多発し得る例として，例えば画像ビューアでは，画像がスライドインし表示される処理，画像の拡大・縮小やスクロール処理などがある．ここでは，ドキュメントビューアで，文章の上にページ番号が重ねて表示される例を説明する．例えば，文章の表示領域 D の書換え命令が発行された直後に，ページ番号を重ねて表示する小さい部分領域 E の書換え命令が発行される場合を考える．

図 2 は，この際に，後発の領域 E の書換え命令が先行す

Vol.2014-SLDM-165 No.29
Vol.2014-EMB-32 No.29
2014/3/16

る領域Dの書換え命令とコリジョンする様子を示している．先行する領域Dの書換え処理が完了すると，EPDコントローラからの割り込みが通知されるので，デバイスドライバは，領域Eの再書換えである E'を指示する．このように，先行する書換え処理が完了するまで後続の書換え処理が待たされるため，書換え処理時間が更に長くなってしまう．

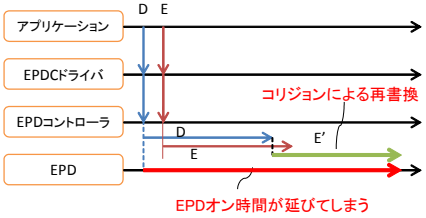

図 2　コリジョン発生による書換え処理時間の増加
Figure 2　Update time increases with collision events.

4. 電子ペーパ書換え処理の低消費電力制御方式

本章では，アプリケーションから断続的に発行される書換え命令から時間的に近接する命令列を検出し，効率よく書換え可能な命令列に再構成する低消費電力制御方式について述べる．

4.1 EPD スケジューラの基本アルゴリズム

3章で示したEPDの消費電力を増加させる問題の回避をすべてのアプリケーションに個別に手を入れて行うのは容易ではない．また，EPD向けにアプリケーションを新規に作成する場合も，EPDの特性を十分に理解しないといけないのでアプリケーション開発者の負担となる．そこで，本稿では，EPDコントローラのデバイスドライバで，EPDに不向きな書換え処理をEPD向けに最適化するEPDスケジューラを提案する．EPDスケジューラは，時間的に近接する書換え命令の実行タイミングを揃えたり，コリジョンを回避することで，EPDやEPDコントローラなどの動作時間を短縮し，EPD搭載端末の超低消費電力化を実現する．

3章で述べた2つの課題について，EPDスケジューラでの解決手法を以下で説明する．

まず，1つ目の課題である集中して発行される複数の書換え命令による動作時間の増加の解決手法について述べる．解決手法では，先行する書換え命令をEPDコントローラにすぐに実行させずに，複数の命令が揃うのを待ってから，書換え開始タイミングを揃えて一気に実行させることで，書換え処理時間の短縮による消費電力の削減を狙う．この際の待ち時間は，例えば数十 ms～100ms 程度と，EPDの書換え時間と比較して十分短くなるようにしてユーザに実行遅延の影響を感じさせないのが望ましい．

図3は，図1の書換え処理に本手法を適用した様子を示している．図に示すように，書換え命令AやBが到着しても，EPDスケジューラではすぐに書換え処理を開始させない．最初の命令Aが到着してから一定時間待ち，その時点

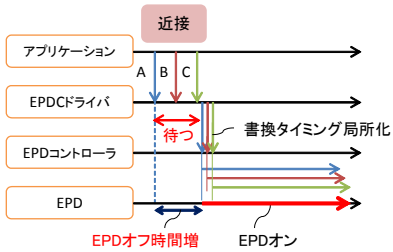

図 3　書換え処理の局所化による低消費電力化
Figure 3　Reducing power with localized updates.

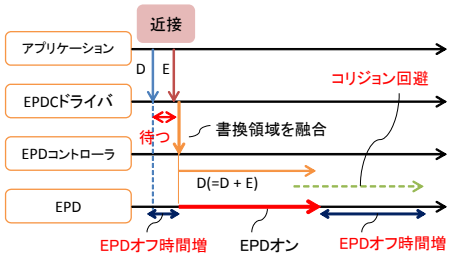

図 4　コリジョン発生回避による低消費電力化
Figure 4　Reducing power with collision avoidance.

でデバイスドライバに到着している命令Cまでを近接書換え命令とみなして，空間的・時間的に再構成してEPDコントローラに書換え処理の開始を指示する．空間的な再構成については，例えば，複数の書換え領域が，ある矩形領域で包含できる場合のみ，同矩形領域を新たな書換え領域としてEPDコントローラに実行を指示する．仮に空間的に再構成できなくても，時間的には再構成できる．複数の書換え処理の開始時間を揃えて可能な限りオーバラップさせる．これにより，EPDやEPDコントローラや主記憶の動作タイミングを局所化し，動作時間を短くできる．

次に，2つ目の課題であるコリジョン発生による書換え処理時間の増加の解決手法を示す．

コリジョンを回避する手法についても，同様に，先行書換え命令の実行を遅延させ，コリジョンを引き起こす複数命令を単一命令に融合してコリジョンを除去することで，書換え処理時間の短縮と消費電力の削減を狙う．

図4は，図2の書換え処理に本手法を適用した様子を示している．図に示すように，最初の書換え命令Dが到着してから一定時間待ち，その時点で溜まっている，コリジョンを引き起こす領域Dと領域Eの書換え命令の単一命令への融合を試みる．この2つの領域は，これを包含する領域Dに空間的に融合可能なので，単一の書換え命令に再構成できる．単一の書換え命令は，当然コリジョンしないのでこれを回避でき，その分書換え処理時間が短縮できる．

これらの手法により，EPD搭載端末のアイドル時間を増やすことができ，不揮発性ディスプレイを採用してアイドル時の消費電力を下げるシステムの省電力性をより引き出すことが可能になる．

Vol.2014-SLDM-165 No.29
Vol.2014-EMB-32 No.29
2014/3/16

4.2 EPD スケジューラの待ち時間の決定方法

EPD スケジューラの基本アルゴリズムは，一定時間待つことで時間的に近接する書換え命令を再構成する．しかし，どの程度待てばよいか決める必要があるので，これについてアプリケーションの観点で議論する．

まず，EPD 搭載端末で想定されるアプリケーションには，PDF リーダや画像ビューアのようなページをめくりながら閲覧するものが多い．これらのアプリケーションでは，ページをめくる毎に，似たような時間間隔で複数書換え命令の発行が繰り返されることが多い．

このような規則性のある書換え命令については，待ち時間をオンラインで自動調整する手法が有効である．本稿では，時間的に近接している書換え命令をグルーピングし，グループ内の命令の発行間隔からどの程度待ったらよいか推定する手法を提案する．

図 5 の具体例を用いて説明する．図では，ページめくり毎に，2 つの書換え命令が繰り返し発行されることを想定している．最初のページめくりで，最初のグループを形成する近接書換え命令 A，B が順次到着する．最初のグループでは，書換え命令 A が到着する 50ms 後には命令 B が到着するが，EPD スケジューラの待ち時間の初期値が 100ms に設定されていると仮定すると，50ms 無駄に待ってしまう．近接書換え命令 C, D が形成する 2 つ目のグループでは，命令 C が到着すると，直前のグループに含まれていた命令例の最初の命令の到着時刻から最後の命令の到着時刻までの時間を新しい待ち時間に設定する．この例では，到着間隔が 50ms だったので，これに繰り返し処理でも間隔がぶれることを考慮した 20ms のマージンを加えた 70ms が待ち時間となり，無駄を小さくできる．このように，書換え命令の発行パタンに応じて動的に待ち時間を自動調整することができる．

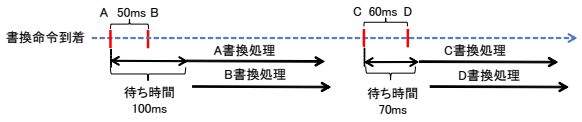

図 5　待ち時間の自動調整手法

Figure 5　Delay time auto-tuning method.

一方で，例えばブラウザでは，表示するページに依って，書換え命令の数や発行間隔も異なってくる．このように規則性がない場合には，一連の書換え命令の最後であることをブラウザからヒント情報として書換え命令に明示的に付加できれば，EPD スケジューラでは最後の命令が到着するまで待ってからまとめて書換え処理でき，余計に待って無駄に電力を使ったり EPD に表示されるまでユーザを余計に待たせなくて済む．

また，ブラウザは，データが揃うまでの比較的長い間，書換え処理が断続的に続くことが多く，最後の書換え命令まで待つとユーザエクスペリエンスを低下させてしまう．そのような場合には，EPD の複数の書換えモードをうまく使えばよい．最初の書換え命令は，すぐに高速モードで書換え処理を行いユーザを待たせないようにする．そして，データが揃った時点で通常モードで 1 回書換えるようなスケジューリングをすれば途中の書換え処理を削減でき，低消費電力化も同時達成できる．

また，アプリケーションに依っては，EPD スケジューラで待たせず，すぐ書換え処理を開始しないといけない場合もある．その最たるものが，ドキュメントワークでは重要となる手書き処理である[12]．例えば，PDF ファイルのページをめくりながら，途中で手書きのメモを残していくユースケースを想定する．この場合，書換え処理を少しでも待たせてしまうと，ペンの動きに書換え処理が追従せず，使い勝手が大幅に低下してしまう．そこで，例えば，書換え領域のサイズに着目する．手書きアプリケーションでは，応答性を向上させるために，手書きされた部分領域の書換え命令を即座に発行するので，書換え領域のサイズが非常に小さいと考えられる．そのため，領域が小さい書換え命令は待たせないようにすれば，手書きの応答性を低下させずに済む．同様に，待ち時間の自動調整の際にも無視すれば，適切な待ち時間の検出に影響しないようにできる．

5. EPD スケジューラの性能評価

EPD スケジューラを，EPD コントローラのデバイスドライバに実装し，性能評価を行った．

5.1 評価環境

評価は，高性能の 800MHz ARM Cortex-A8 と EPD コントローラを内蔵した SoC である i.MX508 アプリケーションプロセッサと 9.7 インチの E Ink EPD 搭載端末評価ボード上で行った．OS は，バージョン 2.3.3 の*Android™ OS を利用した．

5.2 EPD スケジューラの省電力効果の評価

EPD スケジューラの待ち時間をアプリケーションに合わせて設定した場合の低消費電力化効果を検証する．

まず，EPD に不向きな LCD を前提とした書換え処理の例として，*Android™ OS のエフェクトを用いて評価する．

図 6 に，*Android™ OS の Home 画面から Menu 画面へ遷移する際の書換え処理のタイミングの例を示す．図では，書換え命令がデバイスドライバに到着するタイミングと，EPD コントローラで並列に実行される書換え処理（図の矢印）の様子を示している．前述の通り，エフェクトなどは，LCD で操作感を向上させるためのもので，最終的な表示までじわじわと同じ領域を変化させる．そのため，コリジョンが多発し，それに対応する再書換え処理が発生して EPD の書換え処理時間が長くなってしまっている．この結果から，EPD スケジューラで最後の書換え命令の到着まで掛かった時間だけ待てば，コリジョンを回避できることが分かる．

図 7 に，EPD スケジューラを適用した場合を示す．図より，EPD スケジューラで書換え処理を待たせて，1 回の書換え処理に再構成することでコリジョンを回避していることが分かる．その結果，書換え処理時間が 68%削減されている．また，全体の処理時間(最初の書換え命令が到着してから最後の書換え処理が終了するまでの時間)で見ても 28%削減されている．全体の処理時間を削減できている一方で，この例では，ユーザを 1200ms 待たせているのでトレードオフがあり，4.2 節で説明した書換えモードを使い分ける方法で応答性を高めてもよい．EPD 搭載端末が何を重視するべきかは，アプリケーションや電力制約に応じて十分検討されるべきである．

また，前処理や書換え処理を書換え命令が到着したらすぐ行うことをやめたことで，アプリケーションが書換え命令を発行しきるまでの時間も 23%ほど短くなっており，これも省電力化に効いていると考えられる．

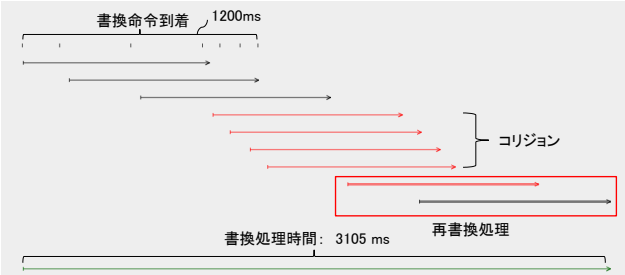

図 6　書換え処理タイミング（LCD 向けエフェクト）
Figure 6　Update timing sequence (Effect for LCD).

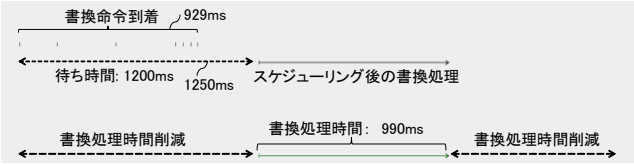

図 7　EPD スケジューラ適用(LCD 向けエフェクト)
Figure 7　Exploiting EPD scheduler (Effect for LCD).

図 8 に，消費電力量削減効果を示す．消費電力量の内訳は，LPDDR2，EPD，ARM コア，EPD コントローラを含むペリフェラル，その他となっている．EPD スケジューラによるコリジョン除去により，EPD 搭載端末全体で 49%の低消費電力化が達成できることが分かった．また，書換え処理時間を短縮したことで，消費電力量が大きかった EPD と EPD コントローラが含まれているペリフェラルの消費電力量の削減が顕著であることが分かる．特に，EPD は 70%削減できている．

次に，PDF リーダを用いて，省電力効果を評価する．図 9 に，PDF リーダでページめくりをした際の書換え処理タイミングの例を示す．評価に用いた PDF リーダでは，書換え命令数も少なく，コリジョンを引き起こす書換え処理も

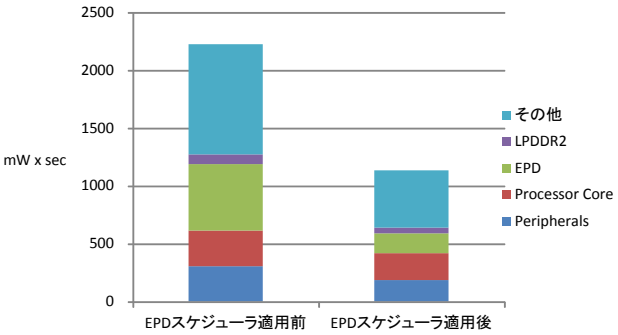

図 8　消費電力量削減効果(LCD 向けエフェクト)
Figure 8　Power consumption (Effect for LCD).

ないので，最適化の余地は LCD 向けエフェクトの場合ほど大きくはない．

図 10 に，EPD スケジューラを適用した場合を示す．EPD スケジューラで書換え処理を待たせて，1 回の書換え処理に再構成している．最初の書換え命令が到着してから，最後の書換え命令の終了時刻までの時間はほとんど変わらないが，その中で書換え処理をしている時間は 13%短くなっている．

図 11 に，消費電力量削減効果を示す．EPD スケジューラを利用することで，複数書換え処理を局所化し，書換え

図 9　書換え処理タイミング（PDF リーダ）
Figure 9　Update timing sequence (PDF Reader).

図 10　EPD スケジューラ適用（PDF リーダ）
Figure 10　Exploiting EPD scheduler (PDF Reader).

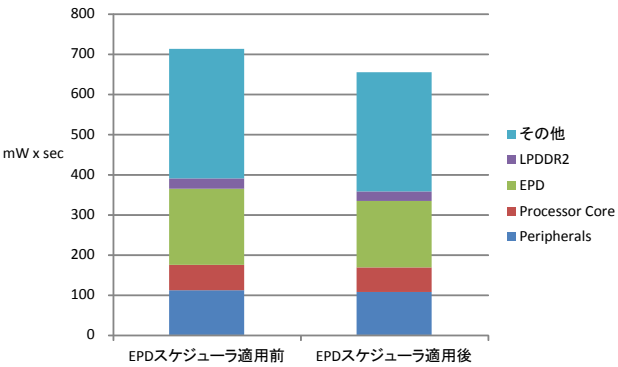

図 11　消費電力量削減効果(PDF リーダ)
Figure 11　Power consumption (PDF reader).

処理時間が削減されたので，EPD の消費電力量削減が 13%となり，全体でも 8%の電力削減となった．最適化の余地が少ない例でも，書換え処理の細かい制御で，うまく電力利用を効率化できることを示唆している．

5.3 待ち時間の自動調整手法の評価

図 12 に，5.2 節で使用した PDF リーダでページめくりを繰り返した際の EPD スケジューラの待ち時間の自動調整の効果を示す．

図では，待ち時間の初期値を 150ms に設定している．最初のページめくりを実行すると，2 つの書換え命令が 101ms 間隔で到着するのでこの時は余計に待ってしまう．次のページをめくると，今度は，先程の 101ms にマージンの 20ms を加えた 121ms で待つようになる．この時は，2 つの書換え命令が 104ms 間隔で到着するので待ち時間は小さくなり，待ち時間が自動調整できていることが分かる．ページによってレンダリング時間にばらつきがあるので，マージンの利用は重要である．

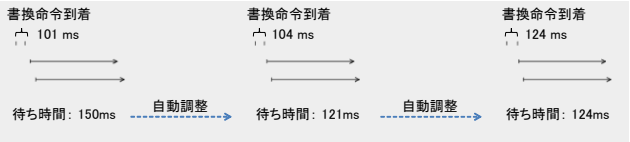

図 12　待ち時間自動調整(PDF リーダ)
Figure 12　Delay time auto-tuning (PDF reader).

6. 今後の課題

本稿では，EPD コントローラのデバイスドライバに EPD スケジューラを実装し，デバイスドライバ単体でできる最適化をおこなった．今後は，これに加えて，アプリケーションに少しだけ手を加えてヒント情報を様々な形式で通知してもらえるようにする．ヒント情報を利用することで，EPD スケジューラの待ち時間をより適切に調整できるようにする．また，待ち時間調整の自動化の適用範囲拡大など EPD スケジューラの機能拡張等を行い，アクティブ時のさらなる低消費電力化を目指す．

また，昨今，本稿で述べたような省電力ディスプレイを利用してアイドル時の消費電力を下げて低消費電力化するシステムに，将来的に活用できそうなディスプレイが EPD 以外にも続々登場している．IGZO などの低リフレッシュレートディスプレイでは，静止画表示のために必要な電力は劇的に小さくなっているものの，EPD 搭載端末で想定しているような厳しい電力バジェットで駆動させることを考え始めると，書換え時の消費電力を徹底的に削減する必要がある．その際に，EPD スケジューラで行っているような最適化技術が，幅広く適用できる可能性がある．具体的には，短い間隔で全面を書換える命令が発行された場合，最後の 1 回だけ書換えるように制御すればよい．LCD のように書換え時間が短いディスプレイでは，仮に 3 つの書換え処理をまとめることができれば，書換え処理に掛かる消費電力を約 1/3 にできる可能性がある．このように，本手法は，書換え時間が短くなるとさらに効果が大きくなることが期待できる．今後は，EPD 以外の省電力ディスプレイ向けの低消費電力制御技術の確立も目指していく．

7. おわりに

不揮発性のディスプレイである EPD を採用してアイドル期間の消費電力を削減するシステムにおいては，アクティブ時の低消費電力化も重要な課題となる．電子ペーパ搭載端末のアクティブ時の消費電力の大部分を占める EPD の書換え処理の低消費電力化を行った．

本稿では，EPD の書換え時間の長さやコリジョンなどの書換え処理時間が長くなる要因に着目し，これらを回避できるように書換え命令をスケジューリングすることで書換え処理時間を削減して低消費電力化する EPD スケジューラを提案した．EPD スケジューラを実装し，EPD 搭載端末で想定されるワークロードを用いて有効性の評価を行った結果，低消費電力化できることが確認できた．

参考文献

1) Bhowmik A. et al.: System-Level Display Power Reduction Technologies for Portable Computing and Communications Devices, IEEE Portable (2007).
2) Han K. et al. : A Hybrid Display Frame Buffer Architecture for Energy Efficient Display Subsystems, International Symposium on Low Power Electronics and Design (ISLPED), pp. 347-352 (2013).
3) 松尾拓哉: IGZO 技術，シャープ技報，第 104 号(2012).
4) 日本画像学会: 電子ペーパ，東京電機大学出版局 (2008).
5) E Ink: http://www.eink.com/
6) Cummings W.: mirasol® -- Revolutionary color display technology for diverse mobile applications, Proc. of International Symposium on Electronic Paper(ISEP), pp.12-16(2012).
7) Watts J.D.: Presenting a 10.7” flexible colour electronic paper display fabricated using a qualified manufacturing process, Proc. of International Symposium on Electronic Paper(ISEP), pp.17-21(2012).
8) Kindle: https://kindle.amazon.com/
9) Toq: https://toq.qualcomm.com/
10) YOTAPHONE: http://yotaphone.com/
11) i.MX50 Multimedia Applications Processor Reference Manual, Chapter 24 Electrophoretic Display Controller, Rev.1, Freescale Semiconductor, Inc. (2011).
12) Nebashi S.: High Resolution E-Paper System Platform, Proc. of International Symposium on Electronic Paper(ISEP), pp.27-30(2012).

*Android は， Google Inc. の商標です．

本論文に掲載の商品，機能等の名称は，それぞれ各社が商標として使用している場合があります．