

SOTBを用いたCPUの省電力OSスケジューラの基本設計

濱田 槇亮^{†1,a)} 小柴 篤史^{†1} 並木 美太郎^{†1}

概要: IoTの実現に向けて低消費電力LSIは不可欠である。LSIを低消費電力化する技術として、Silicon on Thin BOX(SOTB)によるボディバイアス制御が有効であることが知られている。ボディバイアス制御は、最大動作周波数とリーク電力のトレードオフを取ることができ、その特性を計測・評価した先行研究が行われている。しかし、先行研究ではソフトウェアを含めたシステム全体での最適化ボディバイアス制御に関する議論は行われていない。OSが実行中のアプリケーションの特性に合わせた適切なボディバイアス制御を行うことで消費電力を削減することを目標とし、本研究では、その前段階としてSOTB実装CPUの特性の計測と消費電力を削減するOSスケジューラを提案する。ボディバイアス制御によって全体として得となっているかどうかの指標となるBreak Even Timeや、ボディバイアス制御による基盤特性の変化が現れるまでの待機時間の計測をした。OSスケジューラは、スリープ可能な時間に応じて適用するボディバイアス制御の範囲を切り替えることで、より効率的に消費電力の削減を狙う。今後の課題として、提案するスケジューラと制御機構を実装し、実測にて削減効果を評価する。

1. はじめに

Internet of Things(IoT)の実現にむけて低消費電力マイクロコントローラへの需要が高まっている。ウェアラブルデバイスなどのバッテリー駆動による運用が想定される場合では、供給可能な電力に限りがあり、消費電力の削減は必須課題となっている。マイクロコントローラなどのLSIの消費電力を削減する方法として、超低電圧デバイス研究組合(LEAP)が開発したSilicon on Thin Buried Oxide(SOTB)によるボディバイアス制御が有効的である[1]。SOTBを用いて製造されたLSIは、ニアスレシールドコンピューティングによって電源電圧を閾値電圧付近まで下げることができ、電源電圧が0.5V未満での動作が可能となっている。ボディバイアス制御は、基板に印加するバイアス電圧を変化させることによって、LSIのリーク電流量と最大動作周波数のトレードオフを取ることができ、どのようにトレードオフを取るかは、様々な検討が行われている[2][3][4][5]。しかし、実行中のアプリケーションの状況も含めた検討は行われておらず、検討の余地が存在する。

物理的なアプローチによって消費電力の削減を図る方法はボディバイアス制御以外にも存在する。DVFS(Dynamic Voltage and Frequency Scaling)やPG(Power Gating)では、様々な消費電力削減の手法が検討されている。PGは、回路の使用しない部分の電源を切ることで消費電力の削

減を狙うものであり、電源のオンオフの際に電力オーバーヘッドが存在する[7][8][12][10][9][11][13][14]。このオーバーヘッドを上回るようにPG制御する必要があり、この性質はボディバイアス制御にも存在している。PG向けに考案された手法を、動的ボディバイアス制御でも適用できるかは不明であるが、似た性質を持つことから適用できる可能性が存在する。

本研究では、SOTBを適用したCPU「GC-SOTB」を使用する。GC-SOTBは独自に開発したチップであり、GC-SOTBのボディバイアス制御の電力特性など物理的特性は判明していない。本論文では、この特性を解明し、動的ボディバイアス制御をするうえで、どのように活かせるか検討する。さらに、特性に合わせた、より消費電力を削減できる手法を提案し、それを実現するために必要となる新しい制御コントローラも提案する。

2. Silicon on Thin Buried Oxide

SOTBは、超低電圧デバイス研究組合(LEAP)が開発した低電力組み込みシステム向けのSOI(Silicon On Insulator)トランジスタであり、FD-SOI(Fully Depleted-Silicon On Insulator)の一種である。Buried Oxide(BOX)層の上にトランジスタを形成することによって、チャネルへの不純物のドーピングが不要となっており、不純物ゆらぎによる閾値電圧のばらつきが小さくなっている。これにより電源電圧を閾値電圧付近まで下げることが可能となっている。また、BOX層は基板とドレインソース領域間に存在するため、

^{†1} 現在、東京農工大学

^{a)} hamada@namikilab.tuat.ac.jp

リーク電流が発生を抑えることができる。BOX層の下層に位置するウェル領域は、トリプルウェル領域を構成しており、バイアス電圧を印加することによってリーク電流を抑えることができる。先行研究では、リーク電力を89.7%削減できることが確認できている [3]。

2.1 ボディバイアス制御

ウェル領域に印加する電圧 (バイアス電圧) を変更することによって閾値電圧を変更することができる。これをボディバイアス制御 (基板バイアス制御) と呼ぶ。また、LSI に電源電圧を与えて動作しているときに、行われるボディバイアス制御を動的ボディバイアス制御と呼ぶ。

電源電圧を V_{DD} とし、ウェル領域の nMOSFET のボディ端子電圧を V_{BN} 、ウェル領域の pMOSFET のボディ端子電圧を V_{BP} としたとき、

$$GND < V_{BN}, V_{DD} > V_{BP} \quad (1)$$

が成立するときをリバースバイアスと呼ぶ。基板にリバースバイアスがかかっている状態では、閾値電圧が上昇し、リーク電流を削減することができる。しかし、LSI の遅延速度が増大するため最大動作周波数が低下してしまう。

対して

$$GND > V_{BN}, V_{DD} < V_{BP} \quad (2)$$

が成立するときをフォワードバイアスと呼ぶ。フォワードバイアスでは、閾値電圧が低下するため、リーク電流が増加するが、遅延時間が減少される。よって最大動作周波数が上昇することになる。また、ボディとソースの電圧が等しいときをゼロバイアスと呼び、これを標準的なバイアス状態とする。

ボディバイアス制御によって、基板の特性を変更することができる。リーク電流と実行性能のトレードオフを取ることができる。最も典型的な制御方法として、処理性能を必要としない時にはリバースバイアスをかけて待機状態へとスリープさせて、処理性能が必要な時には、フォワードバイアスをかけて実行速度を上げる、といった方法が考えられる。PG と違い、動的ボディバイアス制御ではリバースバイアスを適用してスリープさせてもレジスタやキャッシュ、パイプラインの情報が失われることが、ないため復旧せずに速やかに処理に戻ることが可能である。

2.2 GC-SOTB

GC-SOTB は、共同研究の一環として開発されたチップであり SOTB を用いて開発された。32bit MIPS 準拠アーキテクチャであり、データキャッシュと命令キャッシュをそれぞれ持っている。TLB を内蔵しているため、仮想ページングが可能であり、単一のアプリケーションを実行するマイクロコントローラとしての使用だけでなく、Linux などの

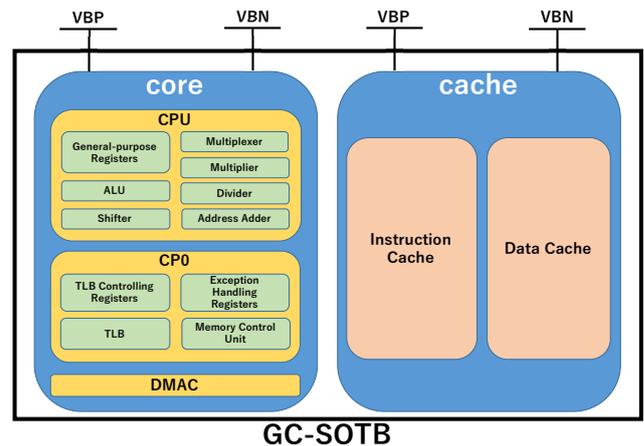


図 1 GC-SOTB 構成

汎用 OS を動作させることが可能となっている。GC-SOTB は、CPU としての機能や TLB などの機能をまとめた cache 部と、命令キャッシュとデータキャッシュから構成される cache 部の 2 つの部分で構成されている (図 1)。core 部と cache 部は、それぞれ独立してボディバイアス制御ができるようになっており、片方のみをスリープさせることも可能である。

GC-SOTB は独自開発であるため実行環境も独自の環境となっている (図 2, 図 3)。GC-SOTB は CPU としての機能しか持たないため、メモリやクロックを持っていない。評価用マザーボードには、GC-SOTB だけでなく FPGA も接続することができる。FPGA 上で RAM や ROM、クロックなどの周辺機器を作成し活用する。また、ボディバイアス制御には異なる電源が複数必要である。GC-SOTB には、core と cache それぞれに V_{DD} 、 V_{BP} 、 V_{BN} が存在するため 6 種類の電源が必要である。複数の電圧を管理し制御するために評価用マザーボードには、電源ボードを複数枚接続することができる。電源ボードは GC-SOTB へと印加する電圧を制御しており、 $-6.0V$ から $6.0V$ の幅で電圧を調整することができる。

3. 関連研究

ボディバイアス制御によってトレードオフを取ることができるため、最適なバイアス状態を検討する研究がおこなわれている。先行研究では、動的ボディバイアス制御を行う上で必要となるパラメータの計測や特性について検討がされている。SOTB を用いて開発された汎用マイクロコントローラの実機を用いて計測しており、リバースバイアスによって動作時のリーク電力を 89.7%削減できることが確認されている [3]。また、ボディバイアス制御による基板の特性変化をモデル化し、式をあてはめることによって最適となる電源電圧とバイアス電圧の組み合わせを導出する研究もおこなわれている [4]。リアルタイムシステム向けのボディバイアス制御の初期検討もおこなわれている [6]。

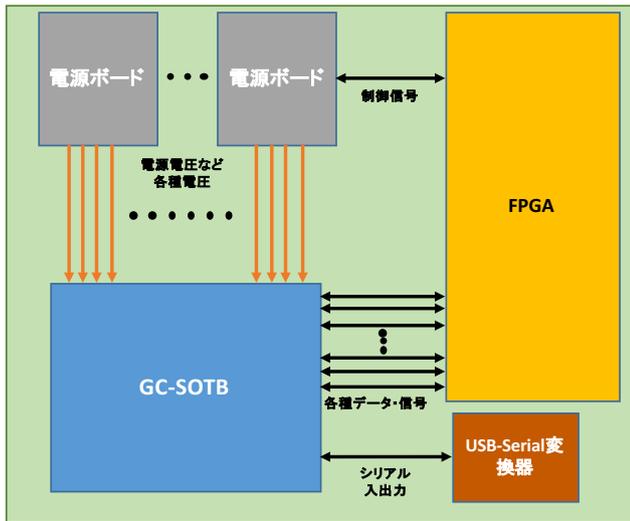


図 2 GC-SOTB 評価環境 構成



図 3 GC-SOTB 評価環境 (左上: 電源ボード, 左下: GC-SOTB, 右下: FPGA ボード)

先行研究では、動作するアプリケーションの状況まで組み込んだバイアスの最適化までは行われていない。また、実際に SOTB を用いて開発されたプロセッサ上でプログラムを実行させながら、動的ボディバイアスを行ったとする報告がない。動的ボディバイアス制御が行える既存の OS は存在しないため、OS による動的ボディバイアス制御は検討もされていない状態となっている。

4. GC-SOTB 基礎評価

GC-SOTB は、SOTB を使用しているため電源電圧の低下に成功している。動作可能な電源電圧の下限値を探るために、電源電圧と動作周波数を変化させながら、GC-SOTB

が動作するか確認した。また、その時に電源電圧から流れる電流値を計測した。確認と計測は、cache と TLB を有効化か無効化した状態の 2 種類の状態でそれぞれ行った。結果は、表 1 と表 2 のとおりとなった。表中の数値は正常に動作したときに流れている電流値を示している。数値でなくが x となっている箇所は、動作しなかったことを示している。cache と TLB を使用しない状態であれば、電源減圧は 0.4V で動作することが可能となっている。また有効となっている場合でも、電源電圧は 0.5V 未満で動作することが確認できている。市販品のマイクロコントローラでは、3.3V や 5.0V で動作するものが多く、半分以下の電源電圧で同等の動作周波数で動作できることが確認できた。GC-SOTB では、0.5V 未満で Linux などの汎用 OS の動作が可能であると言える。

ボディバイアス制御によって基板特性がどのような変化を見せるのかも確認した。電源電圧 0.6V、動作周波数 20MHz で、0.3V のフォワードバイアス、0.3V のリバースバイアス、0.6V のリバースバイアスを適用すると、電流値は表 3 のとおりとなった。static 電流は、クロックを与えない状況で計測した電流値であり、その値はリーク電流によるものが多く占めている。dynamic 電流はクロックを与えた状況で計測した電流値から static 電流の値を引いたものである。演算処理や回路のスイッチングによる電流が多くを占めている。ゼロバイアスを基準として比較すると、リバースバイアスを適用時では static 電流が大きく減少している。0.3V のリバースバイアスでは static 電流が 77% 減少している。0.5V のリバースバイアスでは 91% 減少している。しかし、20MHz での動作ができず、LSI の遅延時間の増加による影響が見られる。対して、フォワードバイアスを適用すると、static 電流が 6 倍になっている。

ボディバイアス制御によって、基板特性に変化が現れることを確認することができた。リバースバイアスを適用することで、リーク電流の削減に成功していることを計測にて確認することができた。

5. Break Even Time

ボディバイアス制御は、バイアス電圧を変化させることによって実行される。電圧を変化させる際に、MOSFET のウェル領域の充放電が発生する。充電に必要なエネルギーは、ボディバイアス制御による電力的なオーバーヘッドとみなすことができる。

電力オーバーヘッドを $E_{overhead}$ 、ボディバイアス制御によって削減できたリーク電流を $i_{leakgain}$ 、電源電圧を V_{DD} とすると、ボディバイアス制御によって削減されたエネルギーと発生したオーバーヘッドが等しくなる時間 Break Even Time(BET) は、以下のように定義される。

$$E_{overhead} = \int_0^{BET} i_{leakgain} V_{DD} dt \quad (3)$$

表 1 電源電圧と動作周波数ごとの動作可否と動作時の電流値 [mA] (cache/TLB off/off)

	8MHz	10MHz	15MHz	20MHz	30MHz	50MHz	70MHz	75MHz	80MHz	100MHz
0.4V	4	×	×	×	×	×	×	×	×	×
0.45V	5	6	10	×	×	×	×	×	×	×
0.5V	6	7	11	14	×	×	×	×	×	×
0.6V	7	9	13	17	26	×	×	×	×	×
0.8V	10	12	19	25	36	59	×	×	×	102
1.0V	13	17	25	32	48	77	101	101	×	×
1.2V	17	21	32	42	61	97	127	127	134	×

表 2 電源電圧と動作周波数ごとの動作可否と動作時の電流値 [mA] (cache/TLB on/on)

	8MHz	10MHz	15MHz	20MHz	30MHz	50MHz	70MHz	75MHz	80MHz	100MHz
0.4V	x	×	×	×	×	×	×	×	×	×
0.45V	5	6	x	×	×	×	×	×	×	×
0.5V	6	7	11	14	×	×	×	×	×	×
0.6V	7	9	13	17	26	×	×	×	×	×
0.8V	10	12	18	24	36	58	×	×	×	x
1.0V	13	16	24	32	48	76	101	100	×	×
1.2V	17	21	31	41	61	97	127	127	x	×

ボディバイアス制御によってリーク電流が削減され始めた時刻を式中的の $t = 0$ としている。

ボディバイアス制御によって消費電力の削減を狙ううえで、BET 未満のリバースバイアスによるスリープは電力的に損をすることになるので、安易なスリープは抑制すべきである。BET 以上の時間だけスリープが行える場合のみリバースバイアスによるスリープを行うことを制御の基本方針とする。制御のためには BET を知ることは重要であり、計測もしくは導出する必要がある。

5.1 BET の測定

BET は、リバースバイアスによって発生する電力オーバーヘッドと削減されたリーク電流量を計測し、これらの値から算出する。削減されたリーク電流量は、ゼロバイアス時の static 電流値とリバースバイアス適用時の static 電流値を計測し、差し引けば求めることができる。電力オーバーヘッドは電流計や電圧計での計測ができないため、オシロスコープを用いて行った。電力オーバーヘッドの計測は電圧プローブと電流プローブを測を行う VBP もしくは VBN を接続し波形を観測した。電圧と電流の積から電力を求めることができる。

ゼロバイアスとリバースバイアスのバイアス状態の遷移にはファンクションジェネレータを用いた。ファンクションジェネレータの出力を接続し、ゼロバイアスと 1.2V のリバースバイアスとなるような電圧をバイアス電圧に印加した。

GC-SOTB は、core と cache で独立して電源電圧とボディバイアス電圧を持っているため、それぞれの値を計測した。

5.2 計測結果

計測の結果から、削減できるリーク電流量は、表 4 のと

おりとなった。cache は単純なメモリセルで構成されているため漏れ出るリーク電流量が core よりも大きくなっている。そのためリバースバイアスによるリーク電流の削減効果も core よりも大きく出ている。GC-SOTB は、core と cache の電源が独立しているため、削減できる電流は単純に足しあわせて考えていい。

オシロスコープによってバイアス状態が変化する前後の状況を観測した結果は、図 4 と図 5 のとおりであった。黄色の波形はバイアス電圧を示しており、青色の波形は、バイアス電圧にかけた電源から流れる電流である。桃色は、バイアス電圧と電流の積をとったものを示しており、波形の凸になっている箇所の面積が電力オーバーヘッドである。電力オーバーヘッドは、表 5 のとおりとなっている。オーバーヘッドは充放電が起きるために生じるものであり、core は CPU 機能を持っているため構造が複雑であり、単純な構造である cache と比較すると静電容量が大きくなると考えられる。そのため、電力オーバーヘッドも core の方が大きくなっていると推察される。

削減できるリーク電流量と電力オーバーヘッドから、BET を算出した結果が表 6 となっている。「core+cache」は、同時にリバースバイアスにかけた場合の BET であり、core と cache のそれぞれの BET の平均となっている。同時に制御する場合には、11ms 以上のリバースバイアスを適用しないと得とにならない。cache 単体でも 2.6ms 以上のリバースバイアスを適用する必要がある。GC-SOTB のボディバイアス制御は、ミリ秒オーダーでのスリープが可能である組み込み機器における使用が有効的となっている。特に間欠制御では、より長期のスリープが期待できる。

6. 遷移時間

バイアス電圧を変更することによってウェルの充放電が

表 3 バイアス状態ごとの電流値 (電源電圧 0.6V, 動作周波数 20MHz)

バイアス	VBN [V]	VBP [V]	static 電流 [uA]	dynamic 電流 [mA]	動作成否	補足
0.5V リバースバイアス	-0.5	1.1	46	16.03	x	20MHz で動作しなかった
0.3V リバースバイアス	-0.3	0.9	117	16.06	o	
ゼロバイアス	0.0	0.6	506	16.43	o	
0.3V フォワードバイアス	0.3	0.3	3350	18.47	o	



図 4 電力オーバーヘッド (VBP)

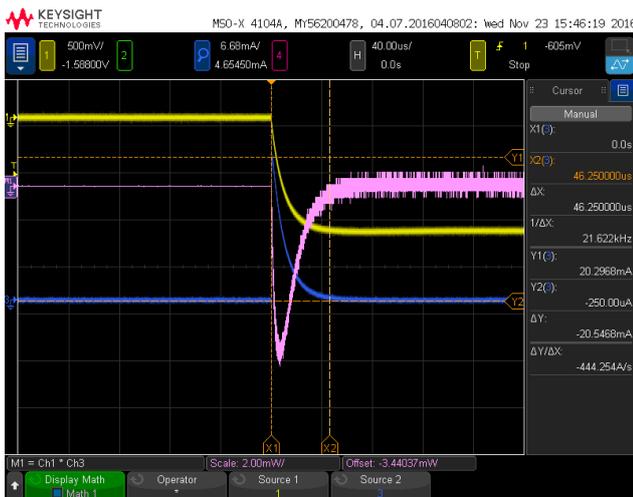


図 5 電力オーバーヘッド (VBN)

表 4 削減されたリーク電流 (リバースバイアス 1.2V)

	Leak gain [uA]
core	90.0
cache	327.0
core + cache	417.0

表 5 電力オーバーヘッド (リバースバイアス 1.2V)

	Energy Overhead [uJ]
core	0.963
cache	0.499
core + cache	1.462

起きる。充放電が始まってから完了するまでの間は、バイアス状態が不定でありシステムとして動作の保障ができな

表 6 Break Even Time (リバースバイアス 1.2V)

	BET [ms]
core	17.837
cache	2.546
core + cache	10.192

い状態である。バイアス電圧を変更してから、基板の特性に変化が現れるまでの遅延をバイアス状態の遷移にかかる時間として、遷移時間と定義する。フォワード方向へのバイアスへと変化させるときの遷移時間を wakeup time、リバース方向へのバイアスへと変化させるときの遷移時間を sleepdown time とする。システムの動作保障を担保するためには、バイアス電圧が変化してから遷移時間が経過するまでシステムは待機する必要がある。特にフォワード方向へのバイアスの変化の際には、基板の特性が変化していないのにクロック速度を上げてしまうと、システムが停止する可能性がある。ボディバイアス制御を行う際には、遷移時間だけの待機時間が必要になってくるので、システムはそれを加味したうえで制御する必要がある。

6.1 計測方法

遷移時間はオシロスコープを用いて、電圧プローブでバイアス電圧を、電流プローブで電源電圧から流れる電流を観測することで求めることができる。電力オーバーヘッドの計測時と同様に、ファンクションジェネレータを用いてバイアス電圧にはゼロバイアスと 1.2V のリバースバイアスとなる電圧を交互に入力する。

6.2 計測結果

オシロスコープにて確認できた波形は、図 6 と図 7 のようになった。電力オーバーヘッドの計測の時と同様で、黄色がバイアス電圧である。青色の波形は、電源電圧から流れる電流を示している。電流が、バイアス電圧の変化後から増減している様子がわかる。

バイアス状態の遷移にかかる時間は、表 7 のとおりとなった。動作の保証を得るために、バイアス電圧を変更してから待ち時間は、wakeup と sleepdown でそれぞれ最長となっているものを採用する。従って、リバースバイアスからゼロバイアスへと復帰する際には、120us 程度待機し、ゼロバイアスからリバースバイアスへとスリープさせる際には、140us 程度待機する。



図 6 遷移時間 sleepdown



図 7 遷移時間 wakeup

表 7 遷移時間 (リバースバイアス 1.2V)

		wakeup [us]	sleepdown [us]
core	VBP	55.69	59.38
	VBN	116.88	136.72
cache	VBP	67.44	74.11
	VBN	55.25	83.67

7. OS スケジューラ的设计

7.1 省電力化戦略

計測した GC-SOTB の BET に基づき、OS のスケジューラでより効率的にリーク電力を削減する手法を提案する。基本方針として、実行できるプロセスがないときにリバースバイアスで CPU をスリープさせて、消費電力を削減する。加えて、スリープ時間の長さに応じて、cache 部のみの制御と、core 部と cache 部の同時制御の 2 種類の制御方式を使い分ける。スリープ時間は、実行できるプロセスが存在せずに演算処理が行われない時間の長さを指し、アプリケーションの明示的なスリープ処理によってスリープ時間

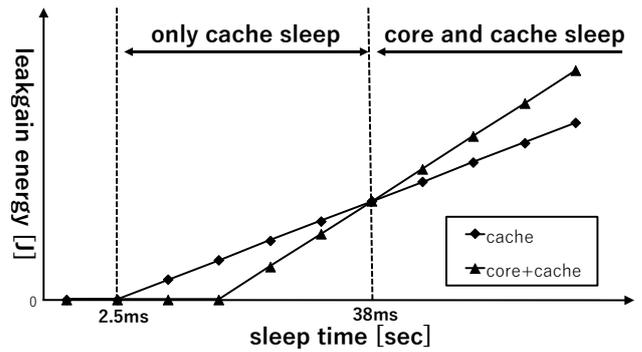


図 8 スリープ時間と削減できるリーク電力量の関係

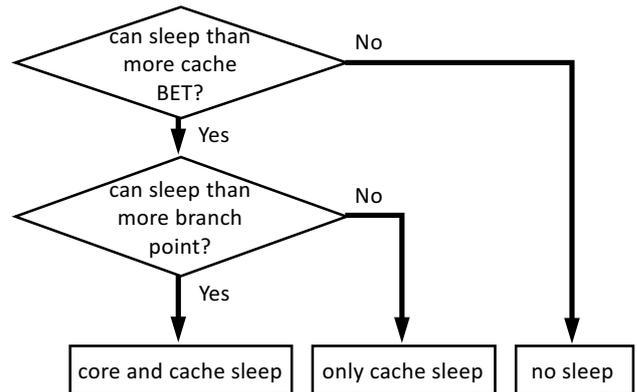


図 9 ボディバイアス制御方式決定の流れ

を算出することができる。

算出された BET と削減できるリーク電流量から、2 つの制御方式での削減リーク電力量の推移を見積もることができる。cache 部のみ制御の方が BET が短く、比較的短いスリープ時間でも消費電力を削減できるが、core 部と cache 部の同時制御の方が時間あたりの削減できるリーク電力が多いため、ある時間からは core 部と cache 部の同時制御の方が得となる。この時間は 38ms となっており、スリープ時間が cache の BET 以上、38ms 未満では cache のみ制御を適用し、38ms 以上では core 部と cache 部の同時制御を適用することで、より効率的に消費電力を削減を狙うことができる。

7.2 ボディバイアス制御コントローラ

リバースバイアスを適用してシステムをスリープさせた場合、スリープ時間経過後にバイアスを元に戻し処理を復帰させる必要がある。core と cache の同時制御では、core をスリープするため CPU とは独立した外部機器からバイアスを制御する必要がある。この外部機器を「ボディバイアス制御コントローラ」と命名し、以下の機能を持つこととなる。

- 電源ボードへと電源制御コマンドの送受信がおこなえる
- CPU へのクロックの供給制御がおこなえる
- 制御レジスタはメモリにマップされており、メモリの

読み書きでCPUから制御が可能である(メモリマップドI/O)

OSは、実行するタスクがなくスリープ時間がリバースバイアスによって得となる時間以上となる判断した場合、次の手順で動的ボディバイアス制御を行う。まずは、実行状態のコンテキストをメモリ上へと退避させる。次に、ボディバイアス制御コントローラへ遷移したいバイアス状態となるバイアス電圧を設定する。設定方法は、制御レジスタがマップされたメモリの書き込みでおこなう。設定が完了したら、制御コントローラを起動する。cacheのみの制御の場合は、待機用命令を実行してスリープ時間だけ待機する。ボディバイアス制御コントローラでは、図10の手順でバイアス電圧の変更などのボディバイアス制御を実行する。

リバースバイアスを適用してスリープさせる際、coreとcacheの同時制御を行う場合ではクロックの停止が必要となる。これは強いリバースバイアスによって動作不可能なクロック周波数となる可能性があるためである。coreとcacheの同時制御が実施する場合は、バイアス電圧を変化させる前にクロックの供給を停止する作業をボディバイアス制御コントローラが行う。cacheのみの制御では、クロックの停止は必要ないため、クロックは供給したままとする。

復帰時の手順は、coreとcacheの同時制御かcacheのみの制御のどちらを実施したかで異なってくる。cacheのみの制御では、バイアス状態が復帰されたら待機用命令を抜けてコンテキストを回復させるだけでよい。この際、バイアス状態が戻ってからwakeup timeだけ待機が必要となる。coreとcacheの同時制御では、coreもスリープさせるため別の方法が必要となる。クロックの供給が再開した後、ボディバイアス制御コントローラは、CPUに対して外部割り込みを発行する。CPUはボディバイアス制御コントローラからの割り込み出会った場合、メモリ上に退避してあったコンテキストを回復することで元の処理に復帰する。

8. おわりに

本研究では、SOTB実装プロセッサ「GC-SOTB」の基礎評価とBET、遷移時間の計測を行った。

GC-SOTBは、0.5V未満で汎用OSを動作させることが可能であり、従来のCPUよりも電源電圧が大幅に下がっていることが確認できた。また、ボディバイアス制御によって基板特性の変化も確認でき、リーク電流を77%削減することが可能であった。

動的ボディバイアス制御をおこなううえで、必要となるBETと遷移時間を計測から求めることができた。

OSがスリープ時間に応じて適用するボディバイアス制御を選択することで、より効率的にリーク電力を削減できる手法を提案した。今後の課題として、提案する手法を実装し実測にてリーク電力削減効果を評価する。

謝辞 本研究はJSPS科研費基盤研究S 25220002及び

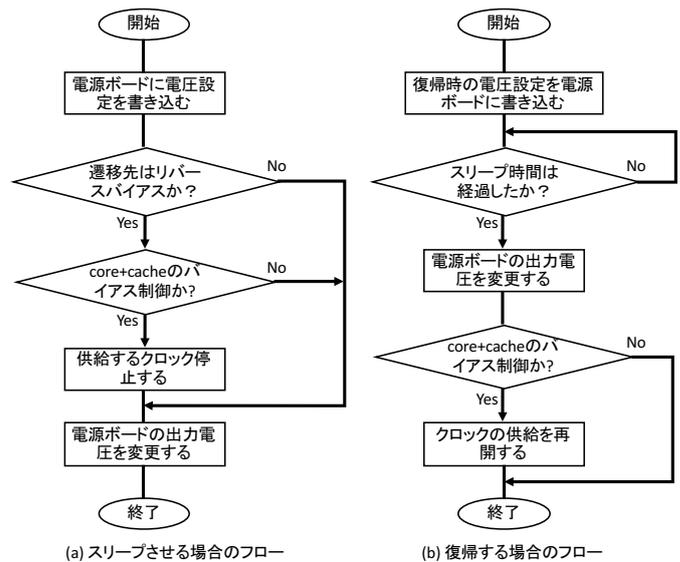


図10 ボディバイアス制御コントローラの動作フロー

特別研究員奨励費 16J06711 の助成を受けたものである。

参考文献

- [1] Takashi Ishigaki, et al. "Ultralow-power LSI Technology with Silicon on Thin Buried Oxide (SOTB) CMOSFET, Solid State Circuits Technologies", Jacobus W. Swart (Ed.), ISBN: 978-953-307-045-2, InTech, pp. 146-156, 2010
- [2] Hayate Okuhara, et al. "A Research of Dynamic Body Bias Control on Micro Controller V850 Using SOTB MOSFET", *IPSJ Journal*, Vol. 57, No.2 Feb.2016, pp. 708-717, 2067.
- [3] 奥原颯, 北森邦明, 宇佐公良, 天野英晴, "SOTB MOSFETを用いた汎用マイクロコントローラ V850 の動的ボディバイアス制御の検討", *情報処理学会論文誌 vol.57 No.2* pp.1-10, 2016-2
- [4] Hayate Okuhara, et al. "An Optimal Power Supply And Body Bias Voltage for an Ultra Low Power Micro-Controller with Silicon on Thin BOX MOSFET", *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2015.
- [5] Hayate Okuhara, et al. "Time Analysis of Applying Back Gate Bias for Reconfigurable Architectures with SOTB MOSFET", *Proceeding of The 19th Workshop on Synthesis And System Integration of Mixed Information Technologies*, pp. 299 - 304 , 2015.
- [6] Carlos C. Cortes Torres, et al. "Analysis of Body Bias Control for Real Time Systems", *The 20th Workshop on Synthesis And System Integration of Mixed Information Technologies*, pp. 48 - 53, 2016.
- [7] Kanak Agarwal, et al. "Power Gating with Multiple Sleep Modes", *ISQED '06 Proceedings of the 7th International Symposium on Quality Electronic Design*, March 27 - 29, 2006, pp. 633-637.
- [8] Hailin Jiang, et al. "Benefits and costs of power-gating technique", *ICCD '05 Proceedings of the 2005 International Conference on Computer Design*, pp. 559-566, 2005.
- [9] Yi-Ping You, C. Lee, and J. K. Lee: "Compilers for leak-age power reduction", *ACM Transactions on Design Automation of Electronic Systems*, Vol. 11, pp. 147?164, 2006.

- [10] A. Shrivastava, D. Kannan, S. Bhardwaj, and S. Vrudhula: “Reducing functional unit power consumption and its variation using leakage sensors” , IEEE Transactions on VLSI Systems, Vol. 18, No. 6, pp. 988-997, 2010.
- [11] S. Roy, N. Ranganathan, and S. Katkooi: “A Framework for Power-Gating Functional Units in Embedded Microprocessors” , IEEE Transaction of VLSI Systems vol.17, pp.1640-1649, Nov. 2009.
- [12] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose: “Microarchitectural techniques for power gating of execution units” , In Proc. of the 2004 International Symposium on Low Power Electronics and design, pp. 32-37, 2004.
- [13] 小林弘明, 佐藤未来子, 天野英晴, 近藤正章, 中村宏, 並木美太郎: “Linux における細粒度パワーゲーティング制御向けコードの実行時管理機構” , 先進的計算基盤システムシンポジウム (SACSIS) 2012, 2012-05-18.
- [14] 小柴 篤史, 塚本 潤, 和田 基, 坂本 龍一, 佐藤 未来子, 小坂 翼, 他: “Linux スケジューラによるリークモニタを用いた細粒度パワーゲーティング制御手法と実チップにおける評価” , 情処研報 Vol.2014-OS-129, No.14, pp.1-9, 2014-05-07.