M101V12

Debug Solutions Power Debugger

Debug System for Boundary Scan Board



Debug Solutions Debug Debug Debug Solutions

ご注意

- 1. このソフトウェアの著作権は、Debug Solutions社にあります。
- 2. このソフトウェアおよびマニュアルの一部または全てを無断で使用、複製することはできません。
- 3. ソフトウェアは、コンピュータ1台につき1セット購入が原則となっております。
- 4. このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用可能です。
- 5. このソフトウェアおよびマニュアルを運用した結果の影響については、いっさい責任をおいかねますので ご了承下さい。

Power Debugger User's Manual Debug Solutions

目次

1. 概要	2
1.1 Power Debugger Ver1.xの概要	
1. 2 Powerデバッガの画面	
	0
2. EDIF <i>JP</i> 1 <i>W</i>	ю
2.1 EDIFファイルについて	6
2.2 EDIFファイルの制限について	6
2.3 EDIFコマンド	7
3. BSDLファイル	
3.1 BSDLファイルについて	9
3.2 BSDLファイルの変換	9
3.3 BSDLファイルの制限	9
3. 4 BSDL⊐マンド	
	19
4. 1 DCFファイルについて	
4.2 DCFファイルコマンド	
5. 試験環境	15
5. 1 試験柔構成	
 5. 試験環境	
 5. 試験環境 5. 1 試験系構成 5. 2 アダプタの設定 5. 3 TRST信号 	
 5. 試験環境	
 5. 試験環境 5. 1 試験系構成 5. 2 アダプタの設定 5. 3 TRST信号 5. 4 EN, XEN信号 5. 5 プリンタポートへの接続 	
 5. 試験環境	15 15 15 16 16 17 18
 5. 試験環境	15 15 15 16 16 16 17 18 19
 5. 試験環境	15 15 15 16 16 16 17 17 18
 5. 試験環境	15 15 15 16 16 17 18 19 20 22
 5. 試験環境	15 15 15 16 16 17 18 19 20 22 26
 5. 試験環境	15 15 15 16 16 17 18 19 20 22 26 31
 5. 試験環境	15 15 15 16 16 17 18 19 20 22 26 31
 5. 試験環境	15 15 15 16 16 17 18 19 20 22 26 31 31 32
 5. 試験環境 5. 1 試験系構成 5. 2 アダプタの設定 5. 3 TRST信号 5. 4 EN, XEN信号 5. 5 プリンタポートへの接続 5. 6 PDA001アダプタのピン配置 5. 7 プリンタポートのピン配置 5. 7 プリンタポートのピン配置 5. 8 PDA002アダプタ 5. 9 PDA012アダプタ 5. 9 PDA012アダプタ 5. 10 PDA013アダプタ 6. バウンダリスキャン試験の流れ 6. 2 プロジェクトの作成と設定 6. 2. 1. プロジェクトの作成 	15 15 15 16 16 17 18 19 20 22 26 31 32 32 32
 5. 試験環境	15 15 15 16 16 17 18 19 20 22 26 31 31 32 32 32 32 32 32

6. 3. 1 Edifファイルの準備	
6. 3. 2 BSDLファイルの準備	
6. 3. 3 DCFファイルへの変換	
6.4 テストパターンの生成	
6. 4. 1 テストパターンの種別	
6. 4. 2 スキャンチェインデバイスのマスク処理	
6. 4. 3 DCF変換フィルタ	
6. 4. 4 EDIF変換フィルタ	
6. 4. 5 テストパターン生成オプション	
6.5 試験の実行	41
6. 5. 1 試験の種類	
6. 5. 2 インフラ試験	
6. 5. 3 ベクタ試験	
6.6 試験結果の解析	
6. 6. 1 インフラ試験	
6. 6. 2 ベクタ試験	
6. 6. 3 結果の厳密なチェック	
7 単体計略	19
7.1 単体試験について	
7.2 BSDLビューにおける設定	
7.3 入力ピンのモニタ	
7.4 出力ピン設定(データ出力後モニタ)	53
7.5 出力ピン設定(モニタ後データ出力)	54
8. オンボードメモリ書き込み	
8 1 オンボードメモリ書き込み冬姓	56
8 1 1 フラッシュメモリ書き込み条件	56
8. 1. 2 FEPROMメモリ書き込み条件	57
8.2 フラッシュメモリ書き込み	59
8.2.1 書き込みファイルのタイプ	59
8. 2. 2 デバイスファイル	
8. 2. 3 チップ構成	
8. 2. 4. 接続信号	
8.2.5 メモリデータ書き込み	
8. 2. 6 書き込み時間の見積もり	
8. 2. 7 書き込み時間の高速化	
8.3 シリアルEEPROM書き込み	
 8. 3. 1 対応デバイス	
8. 3. 2 書き込みファイルのタイプ	
8. 3. 3 デバイスファイル	
8. 3. 4 接続信号	

Power Debugger User's Manual

Debug Solutions

8. 3. 5 メモリデータ書き込み	
9. その他	
9.1 EDIFファイルの階層表示	
9.2 認証コードの登録	

第1章 概要

第1章

概要

Debug Solutions

第1章 概要

1. 概要

1.1 Power Debugger Ver1.x の概要

PowerデバッガはMicrosoft WindowsXP, Windows2000, Windows98 およびWindowsNT 4.0上で動作 するPCBデバッグサポートソフトです。本デバッガは、バウンダリスキャン機能を利用し、PCBの初期不良、デバイスチ ェック、フラッシュメモリの書込み(次期バージョンサポート予定)等を統合的に管理し、PCBのデバッグ効率の向上を はかります。

特徴

・バウンダリスキャン試験テストパターンの自動生成、テスト結果解析機能を標準装備。

・EDIF 200をサポート。

- ・BSDL IEEE1994サポート
- ・バウンダリスキャンチェインに接続されたデバイスの任意入力ピンモニタ/出力ピン設定機能。
- ・他のJtagインターフェース利用アプリとの共存可能なイネーブル信号をサポート。
- ・ターゲットとのインターフェースにプリンタポートを用いているため、ノートブックPCでも利用可能。
- ・EDIFを編集せずに未搭載部品のマスクが可能。
- ・ネット上の任意のポートをバウンダリスキャンポートに指定する仮想ポート機能。
- ・ハードウェアデバッグ時の操作を意識した解りやすいグラフィカルユーザーインターフェース。
- ・WindowsXP, Windows2000, Windows98, WindowsNTに対応。

Microsoft、WindowsXP、Windows2000, Windows98, WindowsNTは米国Microsoft Corporation の米国 及びその他の国における登録商標または商標です。その他、記載されている会社名、製品名は、各社の商標および登 録商標です。

1. 2 Powerデバッガの画面

Powerデバッガの機能として主に4つの画面に分かれています。



図1.2-1 Powerデバッガの画面

(1)Scan Chain View

スキャンチェインリストの作成後、ボード上でのバウンダリスキャンデバイスの接続状態(スキャンチェイン)の表示を 行ないます。表示されているデバイスをクリックすることで、BSDL Viewに対応するデバイス内のバウンダリスキャ ンセル情報を表示します。又デバイス上で右クリックすることでBSDLファイルの表示、DCFファイルの表示、デバイ スの無効化などの処理を行なうことができます。

(2)BSDL View

Scan Chain Viewで選択されたデバイスのバウンダリスキャン情報を表示します。BSDL View内のスキャンチェインセルをクリックすることで対応するピンから出力する値をセットしたり、入力するデータの有効/無効を制御できます。

(3)NetList View

バウンダリスキャンチェインを構成するデバイス間で接続されるネット情報(スキャンネット)が一覧表示されます。ネットリストをダブルクリックすることで対応するネット情報がScan Chain Viewに表示されます。又インフラ試験/

第1章 概要

ベクタ試験後のエラー情報も表示されます。

(4)Command View

Powerデバッガの動作状態を表示します。

エラー発生時コマンドラインをダブルクリックすることで、対応するヘルプを表示します。

第2章 EDIFファイル

第2章

EDIFファイル

2. EDIFファイル

2. 1 EDIFファイルについて

EDIFファイルはElectronic Design Interchange Formatの略でPCB等の配線情報を表すフォーマットです。ED IFのバージョンは現在、1.00から4.00までありますが、バージョン2.00はANSI/EIA Standard 548 - 19 88で標準化されており、世界中のEDAシステムのネットリストや回路図の標準フォーマットして使用されています。

Powerデバッガ はバージョン2.00のEDIFフォーマットのみをサポートしています。PCBの回路図エディタから出力 されるネットリストがEDIF2.00となるように設定してください。

2. 2 EDIFファイルの制限について

Powerデバッガでの、EDIFファイルのファイル名はプロジェクト作成時、 "プロジェクト名.edf" の形式となっていま す。EDIFファイル名をこの形式に変更するか、[ツール] - [プロジェクトパラメータ]で、読み込みEDIFファイルを指定 して下さい。

PowerデバッガのEdifのサポート範囲は以下のとおりになっています。回路図エントリーツールの出力ファイルオプションを以下の形式でEdifファイルが出力されるよう設定してください。

- (1) Powerデバッガはバージョン2. 00 Level 0の EDIFフォーマットのみをサポートしています。
- (2)EDIFのキーワードレベルは0のみです。
- (3)Externalコマンドはサポートしていません。
- (4)階層構造はサポートしていません(平面構造のネットリスト)。
- (5)トップセルの名前はEDIF構造と同じ名前でなければなりません。
- (6)外部ライブラリはサポートしていません。
- (7)cellTypeは "generic" , viewType は "netlist" のみをサポート
- (8)コンポーネントに記述されている内容はBSDL(DCF)のコンポーネント名、ピン名、ピン番号、信号方向が同じで ないといけません。
- (9) array文やportList文は使用できません。

2. 3 EDIFコマンド

コマンドリファレンス付録Aを参照して下さい。

EDIFファイル例:



図2.3-1 EDIFファイル例

第3章 BSDLファイル

第3章

BSDLファイル

Power Debugger User's Manual

3. BSDLファイル

3. 1 BSDLファイルについて

BSDL(Boudary Scan Described Language)はバウンダリスキャン回路の特性を表現したテキストファイルで、 VHDLのサブセットとしてIEEE Std. 1149. 1b-1994(IEEE Standard Test Access Port and Bounda ry-Scan Architecture) に規定されています。

BSDLファイルはデバイスメーカーから取得します。最近ではインターネットのweb上でサポートするメーカーが増えて きました。インストールCDに代表的なデバイスのBSDLファイルが格納されています。データは必ず最新情報を確認 の上御使用下さい。

回路図で使用しているBSDLファイルを "デバイス名.bsd" の形式でプロジェクトのBSDLディレクトリ(¥¥プロジェクトディレクトリ¥BSDL1或いはBSDL2)に保存します。

3.2 BSDLファイルの変換

BSDLファイルはVHDLのサブセットとして規定されており、そのフォーマットはVHDLの表記法と同様な構成となって います。このためピン名の変更等のユーザーによる作業を自動化するのは困難です。そこでPCBデバッガでは、BSD Lフォーマットを単純な構造のDCFファイルに変換することで汎用のテキストエディタやテキスト変換ツール(awk, sed 等)を用いた変更を容易にしています。

BSDLファイルのDCFファイルへの変換はBSDL->DCF変換ダイアログ([部品]-[BSDL->DCF])で行ないます。

3.3 BSDLファイルの制限

(1)STD_1149_1_1990 、STD_1149_1_1993 、STD_1149_1_2001 BSDLに対応。

(2)User Extensions to BSDL 及び User-Supplied VHDL package には未対応

(3) DCFファイルに変換されない構文に関して、文法チェックは行なわれますが、内容に関しては無視されます。

(4)BSDLの各定義文は BSDL description (コマンドリファレンス付録B BSDLコマンド) に示される順で定義 されていなければなりません。(IEEE Std. 1149. 1b-1994)

Debug Solutions

3. 4 BSDLコマンド

コマンドリファレンス 付録B を参照して下さい。

BSDLファイル例:

entity xxxxxxx is Generic parameter Logical port description Use statement use STD_1149_11994. all; Get Std 1149. 1 1194 attribute and definition package STD_1149_1_1994 is	
Give pin mapping declarations Give TAP cntrol declarations	
Give Instruction Register declarations	
—— Give ID and USER code declarations	
Give Register declarations	
Give boudary cell declarations	
—— Boundary Cell deferred cnstants(see package body)	
constant_BC:CELL_INFO;	
<pre>package STD_1149_1_1994 isStandard Boundary Cells constant BC_1:CELL_INFO :=</pre>	
—— Boundary Register declarations	
—— Miscellaneous declarations	
end STD_1149_1_1994;End of 1149.1-1994 Package	
 Device package pin mapping Scan port identification Instruction register description Boundary scan register description 	
end xxxxxxxx	

図3.4-1 BSDLファイル例

Power Debugger User's Manual Debug Solutions

第4章 DCFファイル

第4章

DCFファイル

4. DCFファイル

4. 1 DCFファイルについて

DCF(Device Control Format)ファイルはBSDLファイルの記述を単純化したテキストファイルです。Power Deb ugger は部品情報をDCFの形式で読み取ります。BSDLファイルからDCFファイルへの変換はPower Debugger アプリケーション内の BSDL->DCF変換ダイアログ([部品]-[BSDL->DCF])で行ないます。

〇部品名の一致について

EIDF内のライブラリで定義されている部品名とその部品に対応するDCF内のCompornent文で定義されている内容 が一致していなければなりません。PCBデバッガは、DCFディレクトリ内のDCFファイルを全て読み込み、DCFファイ ル内のコンポーネント名でEDIF内のライブラリに定義されている部品名と結びつけます。

使用している回路図エントリーツールの部品ライブラリがBSDLファイルのコンポーネント名を認識し、出力するEDIFフ ァイル内の部品名がBSDLファイルのコンポーネント名を使用していれば問題ありません。しかし、回路図エントリーツ ールの部品ライブラリが独自のライブラリを使用し、出力するEDIFファイル内の部品名とBSDLファイル内のコンポー ネント名が一致していない場合は、以下のいずれかの作業が必要となります。

(1)回路図エントリーツールの部品ライブラリを変更し部品名をBSDLファイルコンポーネント名と一致させる。

(2) EDIFファイルを編集し、EDIFファイル内の部品名に関する記述をすべてBSDL(DCF)ファイルのコンポーネン ト名と一致させる。

(3)DCFファイルのコンポーネント名を回路図エントリーツールの部品ライブラリと一致させる。

解として(1)の方法ができれば問題ありませんが、ライブラリの変更は相当の作業となり、既存資産との連携もあるため簡単にはできないことが予想されます。

(2)(3)の方法はライブラリの変更なしに、プロジェクト単位での作業が可能です。

(2)のEDIFファイルを編集する場合テキストエディタを用いての手作業の編集が必要となります。EDIFファイル内で回 路図エントリーツールの部品ライブラリで使用されているデバイス名やピン名等の全ての記述をBSDLファイルのコン ポーネント名に変換します。一括変換ファイルを作成し、まとめて変換する方法や、テキストエディタのマクロを使用す る方法もありますが、変換間違い(abcを変換するのにabcdeのabc部分が間違って変換される)に注意する必要があ ります。またEDIF構文の文法の知識も必要となります。

(3)の方法は最も有効な方法といえます。DCFファイルのコンポーネント文の値を回路図エントリーツールの部品名に 合わせる方法です。DCFファイルは "属性(スペース) 値"形式の単純な構文のため属性(ここでは "Compone nt")をキーワードにしてawkコマンドを用いて変換をかければ間違いなく変換が可能です。

さらに簡単な方法としてBSDLファイルを取得してプロジェクトディレクトリのBSDLディレクトリにコピーする際にBSDL のファイル名を"回路図エントリーツールの部品名.bsd"の形式で保存する方法です。この様な形式で保存したBSDL ファイルを[BSDL->DCF]変換ダイアログでDCFファイルに変換する際、『コンポーネント名にBSDLファイル名を 使用する。』オプションをチェックしておくことで回路図エントリーツールの部品名がDCFファイルのコンポーネント名に 使用され、回路図エントリーツールの部品名に対応したDCFファイルが生成されます。(詳細は『コマンドリファレンス 2.3.1章 [BSDL->DCF]コマンド』を参照して下さい)

4. 2 DCFファイルコマンド

コマンドリファレンス 付録C を参照して下さい。

DCFファイル例:

 	このファ Copyrig Tab = 4	イルは P ht Debug を推奨	C B デバ・ Solutions	ッガによっ 3	って作成されました	o			
Target SourceB BsdlUpd BsdlSize NumOff	sdl late Pin	sn74bct8 E:¥Jtag 9574319 4411 24	3244 ¥Jtag¥Sa 92	ample¥Bs	dlNet¥BCT8244.	bsd			
11	Package	Pin Defi	nition						
// // Pin Pin	No. 1 2	Pin_Nar "1GN" "1Y1"	n	Property IN OUTZ	7				
:	÷	÷		÷					
Pin Pin	$23 \\ 24$	"1A1" "2GN"		IN IN					
TCK TDi TDo TMS	13 14 11 12								
// // Instru	ction Reg	ister Defi	nition						
InstrLer InstCap	ngth ture		8 1000000	1					
InstExt InstSam InstByp	est 1pleP 2ass		0000000 0000003 1111111	00 10 1					
 	Scan Cel No.	l Definiti Cell Name	on Pin Name	Pin Num	Attrib	Safe Bit	Cont CellNun	Val to nDisable	Disable Value
Cell Cell	$\begin{array}{c} 0 \\ 1 \end{array}$	$\begin{array}{c} \mathrm{BC}_1\\ \mathrm{BC}_1 \end{array}$	"2Y4" "2Y3"	10 9	Output3 Output3	X X	$\begin{array}{c} 16 \\ 16 \end{array}$	1 1	Z Z
: Cell Cell Cell	$15 \\ 16 \\ 17$: BC_1 BC_1 BC_1	: "1Å1" "2GN" "1GN"		: Input Input Input	: X X X X	: - *	-	- - -
BSRegL	ength		18						

図4.2-1 DCFファイル例

第5章 試験環境

第5章

試験環境

Power Debugger User's Manual

5. 試験環境

5.1 試験系構成

Power DebuggerはIBM PC互換機にインストールして使用します。ターゲットボードとはプリンタポートに接続され たアダプタからケーブルを使用して接続します。



図5.1-1 試験系構成図

※Biosの設定:ターゲットボードへのインターフェースはPCのプリンタポートを使用するため、PCのプリンタポートを有効にする必要があります。PCのBiosの設定でパラレルポートを有効にして下さい。

5.2 アダプタの設定

標準添付のPDA001アダプタを使用時はデフォルトの設定で御使用下さい。

	信号	方向	デフォルト	割当信号				
1	TCK	出力[PC → Targetボード]	D1	DO~D7の信号のいずれかに割りつけます。				
2	TMS	出力[PC → Targetボード]	D3	DO~D7の信号のいずれかに割りつけます。				
3	TDIo	出力[PC → Targetボード]	D5	DO~D7の信号のいずれかに割りつけます。				
4	TDOi	入力[Targetボード → PC]	-BUSY	-BUSY, -ACK, PE, SELECT, -ERR				
				ORの信号のいずれかに割りつけます。				
5	TRST	出力[PC → Targetボード]	D7	DO~D7の信号のいずれかに割りつけます。				
6	EN	出力[PC → Targetボード]	* ¹	DO~D7の信号のいずれかに割りつけます。				
7	XEN	出力[PC → Targetボード]	_ * ¹	DO~D7の信号のいずれかに割りつけます。				
	. 0.							

表5.2-1 アダプタの設定

*1:オプションです。PDA0001には割りつけることができません。フラットケーブルで御使用下さい。

バウンダリスキャン試験を行なう5本の信号(TCK, TMS, TDOi, TDIo, TRST)はプリンタポートから入出力を行いま す。これらの信号をプリンタポートのどの信号に割りつけるかを設定します。信号の割付にはメニューの[ツール]ー[ア ダプタの設定]を選択します。(PDA001使用時)

プリンタポートアドレスはデフォルトで現状使用されているI/Oアドレスが設定されますが、複数のプリンタポートが割りつけられている場合や、異なるポートアドレスを使用する場合はマニュアルで設定して下さい。

O/SにWindowsNTを使用している場合には、専用のプリンタポートI/Oドライバ(PDPortIO. sys)が必要です。P DPortIO. sysファイルはPDPortIOドライバのインストール時にシステムドライブ: ¥ ¥ WINNT ¥ System 32 ¥ Drive rs の下にコピーされ、設定が行われます。

5.3 TRST信号

TRST信号はデバイス内のバウンダリスキャン回路を初期化するための非同期リセット信号です。通常"L"レベルを入 カすることで、バウンダリスキャン回路をリセットし、TEST LOGIC RESETシーケンスに移行します。この状態で は内部論理回路と外部ピンが接続された状態となり、デバイスの通常動作が行なえるようになります。

TCKの5回以上のTMS="1"の入力でTEST LOGIC RESETに遷移することから、TRST信号はオプションとな っており、ターゲットボードにTRST信号を必要とするデバイスが存在する場合に使用します。

Power DebuggerではTRST信号が有効な場合、TRST信号のパルス幅はTEST LOGIC RESETステートの有 効間隔となります。TRST信号有効/無効設定は [ツール]-[アダプタの設定] LPTポートタブで、リセット幅は、 [ツール]-[オプション] 生成&試験タブで行ないます。

5.4 EN, XEN信号

EN, XEN信号はPower DebuggerがJtagポートを使用しているか、使用していないかを示す信号です。Power De bugger以外にJtagを使用するツールを共用して使用することができます。

表 5.4 1 EN, XE	Nの 論 埋	
状態	EN	XEN
Power Debuggerが試験実行中	Н	L
Power Debuggerがアイドル	L	Н

例えば、CPUを搭載するボードであれば、ICEを使用してデバッグすることがあります。Jtagポートが一つであれば、P ower Debugger と ICE それぞれ使用する際にケーブルを付け替える必要があります。しかし、実装が許せば、ボ ードにJtagポートを2個設け、Power DebuggerのEN, XENを使用することでケーブルの差し替え無しにデバッグを 進めることが可能となります。以下にその回路例を示します。



図5.4-1 他のバウンダリスキャンツールとの混在回路例

EN, XENの設定は[ツール]-[アダプタの設定] LPTポートタブ行います。

5.5 プリンタポートへの接続

プリンタポートとターゲットになるPCBは添付のPDA001アダプタを介して接続します。コネクタは物理的な配線のみ で、レベル変換やバッファは入っていません。又パソコンのプリンタポートはTTL5Vインターフェースなので、3.3Vと のインターフェースする場合は注意が必要です。又バッファリングをしているわけではないので、配線長はできるだけ短 くする必要があります。配線を長くしたい場合や電気的インターフェースを変換する場合はPCとターゲットPCBの間に 自作コネクタを使用してください。

5.6 PDAOO1アダプタのピン配置

PDA001はプリンタポートのDSUB25コネクタをRJ45コネクタに変換するためのアダプタです。内部の信号はバッファリングされておらず、単純に接続されているだけです。



図5.6-1 PDA001アダプタ外観図

(1)プリンタポート側[DSUB25-オス]

コネクタのピン番号の並びがPCのプリンタポートコネクタと異なるため、ピン番号のに対する信号名称の並びが異なっています。



試験の実行の前に SELECT(13) 信号と * AUTO(14) 信号の導通チェックを行い、ア ダプタの接続確認を行います。(オプション)

図5.6-2 PDA001プリンタポート側ピン配置

ピン番号	信号名称	接続先*1	信号	ピン番号	信号名称	接続先*1	信号
1	N. C.		-	14	* AUTO	• (13)	内部接続
2	N. C.			15	N. C.	_	—
3	D1	1	тск	16	N. C.		—
4	N. C.			17	N. C.		_
5	D3	3	TMS	18	N. C.	_	_
6	N. C.			19	GND	2	GND
7	D5	4	TDIo	20	N. C.	-	_
8	N. C			21	N. C.		_
9	D7	5	TRST	22	N. C.		_
10	N. C	_		23	GND	6	GND
11	BUSY	7	TDOi	24	N. C.	—	—
12	N. C.	-		25	GND	8	GND
13	SELECT	(14)	内部接続				
*1:ターゲ	ットボード接続	売側のRJ45=	コネクタのピン	番号です。	但し13、14ピ	ンは内部でシ	ョートされてい

表5.6-1 PDA001プリンタポート側ピン配置

* 1:ターゲットボード接続側のRJ45コネクタのピン番号です。但し13、14ピンは内部でショートされてい ます。 (2)ターゲットボード側 [RJ45]



表5.6-2 ターゲットボード側[RJ45]ピン配置

ピン番号	信号	接続先*1
1	тск	D1(03)
2	GND	GND(19)
3	TMS	D3(05)
4	TDIo	D5(07)
5	TRST	D7(9)
6	GND	GND(23)
7	TDOi	BUSY(11)
8	GND	GND(25)
* 1 :プリン	タポート側のI	DSUB25コネクタのI

ン名称です。()内はピン番号です。

図5.6-3 ターゲットボード側 [RJ45] ピン配置

5.7 プリンタポートのピン配置

IBM PC互換PCのプリンタポートを以下に示します。プリンタポートコネクタのピン番号の並びがPDA001アダプタのと異なるため、ピン番号のに対する信号名称の並びが異なっています。



図5.7-1 プリンタポートのピン配置

Debug Solutions

Power Debugger User's Manual

5.8 PDA0027ダプタ

PDA002はUSBインターフェースを用いたポートアダプタです。PDA002アダプタの特徴を以下に示します。

USB側:

•USB1, 1

I/Oポート側:

•3.3Vインターフェース(5Vトレラント)

※Cypress社製EZ-USB(AN2131SC)からの信号を直接引き出しています。電気的インターフェースは、EZ -USB(AN2131SC)のデータシート(http://www.cypress.com/)「Chapter13 EZ-USB AC/DC Pa rameters」を参照願います。

サイズ

19mm × 43mm × 18mm (縦/横/高さ) 突起部含まず

											▼		
	A13	A12	A11	A10	A9	A8	Α7	A6	Α5	A4	A3	A2	A1
	B13	B12	B11	B10	В9	B8	В7	В6	В5	В4	В3	B2	В1
рн 													

図5.8-1 PDA002アダプタのピン配置

番号	信号	IO	論理	内容
A 1	Gnd			GND信号
B 1	Gnd			GND信号
A 2	Vusb(+5V)			USBインターフェー ス5V出 力
B 2	Vusb(+5V)	-	-	USBインターフェー ス5V出 力
A 3	N.C.	1	1	N. C.
В3	Vcc3.3	1	1	3.3 V 出 力
A 4	SDA	1	1	使用不可
B 4	SCL	_	_	使用不可
A 5	PC7	I	正	汎 用 入 力 ポ ート(TD o i) * ¹
B 5	PB0	0	正	汎 用 出 力 ポ ート
A 6	PC6	Ι	正	汎 用 入 力 ポ ート
B 6	P B 1	0	正	汎 用 出 力 ポ ート (T D io) * ¹
A 7	P C 5	I	正	汎 用 入 力 ポ ート
В7	P B 2	0	Ш	汎 用 出 力 ポ ート (TMS) * 1
A 8	P C 4	Ι	正	汎 用 入 力 ポ ート
B 8	P B 3	0	Ш	汎 用 出 力 ポ ート(T C K) * 1
A 9	PC3	Ι	Ш	汎 用 入 力 ポ ート
В9	P B 4	0	正	汎 用 出 力 ポ ート
A 1 0	PC2	Ι	Ш	汎 用 入 力 ポ ート
B10	P B 5	0	ㅂ	汎 用 出 力 ポ ート
A 1 1	PC1	Ι	н	汎 用 入 力 ポ ート
B11	PB6	0	正	汎 用 出 カ ポ ー ト
A 1 2	PC0	Ι	正	汎 用 入 カ ポ ート
B 1 2	P B 7	0	正	汎 用 出 力 ポ ート
A 1 3	PA5	—	—	使用不可
B13	PA4	—	—	使用不可

表 5.8-1 PDA002のピン配置

* 1 : デフォルト設 定

コネクタ

PDA002アダプタ側コネクタ: ヒロセ電機 HIF3F-26PA-2.54DSA

ケーブル側適応コネクタ: ヒロセ電機 HIF3BA-26DA-2.54R

5. 9 PDA0127ダプタ

PDA012はUSB2.0インターフェースを用いたポートアダプタです。PDA012アダプタの特徴を以下に示します。

インターフェース USB側: USB2. 0

I/Oポート側:

3.3Vインターフェース(5Vトレラント)

※FPGA(XC2S100-5TQ144C XILINX)からの信号を直接引き出しています。電気的インターフェースは、XC2S100の データシート(http://direct.xilinx.com/bvdocs/publications/ds001.pdf)を参照願います

サイズ

58mm × 77mm × 31mm (縦/横/高さ) 突起部含まず

ピン配置

В1	В2	В3	В4	В5	B6	В7	B8	В9	в10	
A1	A2	A3	A4	A5	A6	Α7	A8	A9	A10	

図5.9-1 PDA012アダプタのピン配置

番号	ケーブル ピン番 号	信号	IO	論理	内容
A 1	1	LoopOut	0	ㅂ	ケーブルチェック*1
B 1	2	Vcc3.3	—	—	3.3V
A 2	3	DataOut0	0	ㅂ	汎 用 出 力 ポ ート(TCK)
B 2	4	Gnd	-	-	G N D 信 号
A 3	5	DataOut1	0	正	汎 用 出 力 ポ ート (TMS)
В3	6	Gnd	-	-	GND信 号
A 4	7	DataOut2	0	正	汎 用 出 カ ポ ー ト (T D io)
В4	8	Gnd	-	-	GND信 号
A 5	9	DataOut3	0	н	汎 用 出 カ ポ ート(TRST)
B 5	10	Gnd	_	-	GND信 号
A 6	11	DataIn	Ι	н	TD o i 信 号 入 力 ポ ート
B 6	12	Gnd	—	—	GND信号
Α7	13	N .C .	—	—	Reserve信 号
В7	14	Gnd	—	-	GND信 号
A 8	15	DataOut4	0	正	汎 用 出 力 ポ ート(TEN)
B 8	16	Gnd	—	—	GND信 号
A 9	17	DataOut5	0	負	汎 用 出 カ ポ ー ト (X T E N)
В9	18	Gnd	_	_	GND信 号
A 1 0	19	LoopIn	Ι	正	ケーブル チェック*1
B10	20	Gnd	—	—	G N D 信 号

表 5.9-1 PDAO12のピン配置

*1:ケーブルチェック信号。ハード的にA1(ケーブルピン番号1)とA10信号(ケーブルピン番号19)をショートしておく ことで、ケーブル接続チェック機能を使用することが可能となります。

コネクタ

PDA012アダプタ側コネクタ: ヒロセ電機 HIF3BA-20PA-2.54DS

ケーブル側適応コネクタ: ヒロセ電機 HIF3BA-20DA-2.54R

性能

PDA012アダプタは、USB2.0インターフェース使用時で、JTAGポートデータ論理転送レートは、最大1.875MHz となります。又物理速度は、3.75MHzとなります。

LEDの説明



LED1	:	ダウンロード未表示(消灯:ダウ	ンロード完了、点灯∶ダウンロード未)
LED2	:	Power状態表示	(消灯:5V断 、点灯:5V供給)
LED3	:	バウンダリスキャンポートAcce	ss状態表示
			(消灯:アクセスなし、点滅:アクセス中)
LED4	:	未使用	

ケーブルチェック制御

ターゲットボードでA1(ケーブルピン番号1)とA10信号(ケーブルピン番号19)をショートしておくことで、バウンダリス キャン試験時にケーブル接続チェック機能を使用することが可能となります



注意

※PDA012アダプタはバスパワーから、300mA(データ転送時)を必要とします。PDA012アダプタを接続するUSB ポートには、充分な電流供給能力があるポートを使用してください。(PCのUSBポートに直接接続することを推奨し ます。)

※ターゲットボードと接続するケーブルは、必ずターゲットボードの電源をオフした状態で抜き差しを行ってください。

5. 10 PDA0137ダプタ

PDA013はUSB2.0インターフェースを用いたポートアダプタです。PDA013アダプタの特徴を以下に示します。

インターフェース USB側: USB2. 0

I/Oポート側:

2. 5Vもしくは3. 3Vインターフェース(5Vトレラント)

※バウンダリスキャン信号は、SN74ALVTH16244 でバッファリングされています。電気的インターフェースは、TI社 SN74ALVTH16244のデータシートを参照願います

サイズ

90.4mm × 50mm × 26mm (縦/横/高さ) 突起部含まず

ピン配置

A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
B10	В9	B8	B7	B6	B5	В4	В3	B2	B1
μ			I	1	L	1			1

図5.10-1 PDA013アダプタのピン配置

番号	ケーブル ピン番 号	信号	IO	論理	内容
A 1	1	LoopOut	0	н	ケーブル チェック*1
B 1	2	Vcc3.3	-	1	3.3V
A 2	3	Data Out 0	0	н	汎 用 出 力 ポ ート(TCK)
B 2	4	Gnd	—	1	GND信 号
A 3	5	DataOut1	0	正	汎 用 出 力 ポ ート(TMS)
В3	6	Gnd	-	1	GND信号
A 4	7	DataOut2	0	正	汎 用 出 カ ポ ー ト (T D io)
B 4	8	Gnd	-	-	GND信号
A 5	9	DataOut3	0	正	汎 用 出 カ ポ ー ト (T R S T)
B 5	10	Gnd	-	-	G N D 信 号
A 6	11	DataIn	Ι	н	TD o i信 号 入 力 ポ ート
B 6	12	Gnd	-	Ι	GND信号
Α7	13	Vcc2.5V	—	-	2.5V
В7	14	Gnd	_	-	G N D 信 号
A 8	15	DataOut4	0	Е	汎 用 出 カ ポ ー ト (T E N)
B 8	16	Gnd	-	1	GND信 号
A 9	17	DataOut5	0	負	汎 用 出 カ ポ ー ト (X T E N)
В9	18	Gnd	_	-	GND信号
A 1 0	19	LoopIn	Ι	正	ケーブルチェック*1
B10	20	Gnd	_	_	G N D 信 号

表 5.9-1 PDA013のピン配置

*1:ケーブルチェック信号。ハード的にA1(ケーブルピン番号1)とA10信号(ケーブルピン番号19)をショートし ておくことで、ケーブル接続チェック機能を使用することが可能となります。

コネクタ

PDA013アダプタ側コネクタ: ヒロセ電機 HIF3BA-20PA-2.54DS

ケーブル側適応コネクタ: ヒロセ電機 HIF3BA-20DA-2.54R

性能

USBコントローラと内蔵FPGA間のCLK速度は30MHzと48MHzの2種類から選択可能です。このCLK速度により 物理転送性能が決まります。CLK速度が30MHzの時、物理転送性能は7.5MHzとなり、CLK速度が48MHzの時、 物理転送性能は12MHzとなります。最大論理データ転送レートは、物理転送性能の半分となります。実際の転送レー トは、プロトコルオーバヘッドやソフト処理のオーバヘッドのため、最大論理データ転送レートよりも小さい値になりま す。

USBコントローラとFPGA間のCLK速度	物理転送レート	最大論理データ転送レート		
30MHz	7. 5MHz	3. 75MHz		
48MHz	12MHz	6MHz		

48MHz CLKを選択した場合、物理転送速度は12MHzとなります。PDA013アダプタとターゲットボードを接続する ケーブルはできるだけ、短くするようにしてください。



LEDの説明

Access	:	バウンダリスキャンポートAc	cess状態表示 (消灯:アクセスなし 、点滅:アクセス中)
DownLoad	:	ダウンロード未状態表示	(消灯:ダウンロード完了、点灯:ダウンロード未)
Power	:	Power状態表示	(消灯:5V断 、点灯:5V供給)

ケーブルチェック制御

ターゲットボードでA1(ケーブルピン番号1)とA10信号(ケーブルピン番号19)をショートしておくことで、バウンダリス キャン試験時にケーブル接続チェック機能を使用することが可能となります



注意

※PDA013アダプタはバスパワーから、300mA(データ転送時)を必要とします。PDA013アダプタを接続するUSB ポートには、充分な電流供給能力があるポートを使用してください。(PCのUSBポートに直接接続することを推奨し ます。)

※ターゲットボードと接続するケーブルは、必ずターゲットボードの電源をオフした状態で抜き差しを行ってください。

第6章 バウンダリスキャン試験

第6章

バウンダリスキャン試験
6. バウンダリスキャン試験

6. 1 バウンダリスキャン試験の流れ

バウンダリスキャン試験のフローを以下に示します。





6. 2 プロジェクトの作成と設定

6. 2. 1. プロジェクトの作成

Power デバッガの初期立ち上げ画面(メインフレーム)からまずプロジェクトを作成します。プロジェクトの作成にはメ インフレームメニューの[ファイル]ー[新規プロジェクトの作成] を選択します。デフォルトでは、プロジェクトディレクトリ の下にBSDL1ディレクトリ、BSDL2ディレクトリとDCFディレクトリを作成します。

(プロジェクト)ディレクトリ	(BSDL1)ディレクトリ	:ローカルなBSDLファイルを格納します。
	(BSDL2)ディレクトリ	:共通的なBSDLファイルを格納します。
	(DCF)ディレクトリ	:BSDLファイルを変換したDCFファイルを格納します。
	プロジェクト.edf	:回路図入力したCADから出力されたeidfファイル

※これらのディレクトリはデフォルトの名前で作成され、プロジェクト作成後は、[ツール]-[プロジェクトパラメータ]の 『プロジェクトの設定ダイアログ』から変更可能です。

※BSDL1ディレクトリにはプロジェクト固有のBSDLファイルを格納し、BSDL2ディレクトリにはプロジェクト間で共通 なBSDLファイルを格納することで効率的な管理が可能となります。

※DCFディレクトリは、『BSDL->DCF変換』でBSDLを変換したDCFファイルが格納されるディレクトリです。 ※回路図入力したCADから出力したeidfファイルは(プロジェクト)ディレクトリの直下に保存します。

6. 2. 2. プロジェクトの設定

プロジェクトの作成後、プロジェクト固有の情報を [ツール]-[プロジェクトパラメータ] から設定します。

(1)インターフェースポート

インターフェースポートはターゲットボード上のJtagポートのポート名です。ターゲットボードのEdifファイルでトップ セルのポートで定義されているJtagポート(TCK, TMS, TDi, TDo)信号名称を設定します。

通常、Jtagポートはコネクタに割り当てられるため、EDAツールによっては、出力されるEDIFファイルのJtagポートがトップセルのポートとならない場合があります。このような場合、仮想ポート設定を行なうことで回路上の任意のポートをインターフェースポートに割り当てることが可能となります。

(2)仮想ポート設定

仮想ポートとはターゲットボード上の任意のネットをバウンダリスキャンチェインの入出カポートとして扱うものです。 バウンダリスキャンチェインの入出カポートがコネクタに割り当てられていない場合やEDIFのポートに記述されて いない場合、この設定を行うことで、簡単にインターフェースポートの設定が可能となります。

仮想ポートの設定は [ツール]ー[オプション] 仮想ポートタブで行います。部品インスタンス名及びポートインスタンス名の設定はEDIFファイルの読み込み以降に有効となります。

6.3 試験準備

6. 3. 1 Edifファイルの準備

CADソフトから出力されたPCBの回路図情報であるEdifファイルを¥¥プロジェクト名ディレクトリに保存します。デフォルトでは、EDIFファイル名は、プロジェクト名.edfの形式です。[ツール]-[プロジェクトパラメータ]で変更可能です。

6. 3. 2 BSDLファイルの準備

デバッグを開始する前にPCBに搭載されている部品のBSDLファイルを入手する必要があります。BSDLファイルは

¥¥プロジェクト名¥¥BSDL1

のディレクトリ、或いは

¥¥プロジェクト名¥¥BSDL2

のディレクトリの下に保存します。

(BSDL1ディレクトリにはプロジェクト固有のBSDLファイルを、BSDL2には共通的な部品のBSDLファイルを格納すると管理が簡単になります。)

6.3.3 DCFファイルへの変換

BSDLファイルをPCBデバッガ部品フォーマットのDCFファイルに変換します。[部品]-[BSDL->DCF]を選択後、 『BSDL->DCF変換ダイアログ』から変換を行ないます。

DCFファイルは "属性 (スペース) 値"形式の単純なテキストファイルの為、汎用のテキストエディタやテキスト変換ツール(awk, sed等)を用いて変更可能です。

回路図エディタの部品ライブラリで使用されている値(部品名やポート名)がBSDLファイルと異なる場合、回路図エディタから出力されたEDIFファイルとBSDLファイルの整合がとれません。この場合、EDIFファイルを編集してBSDLの 値に合わせる方法と、BSDLファイルを編集して回路図エディタの部品ライブラリと合わせる方法の2通りが考えられま す。

EDIFファイルを編集する方法は使用している部品が少数の場合は問題ありませんが、数が多くなるとその変更量は 膨大なものとなります。BSDLファイルを変更する場合もBSDLファイルはVHDLライクなフォーマットの為、その変更 も単純な置き換えでは不可能です。

この為PCBデバッガでは、BSDLファイルを単純なDCFファイルに変換し、このDCFファイルを用いることにより柔軟

Debug Solutions

性を確保しています。DCFファイルは "属性 (スペース) 値" の単純なフォーマットのなので、属性をキーワードに して、テキスト変換ツール(awk, sed等)を用いて簡単に、回路図エディタで使用している部品ライブラリのデータ値に 変更可能です。(変更はユーザーでプログラムする必要があります。)

又、DCF変換フィルタを用いれば、DCFファイルを読み込んだ後でもDCFファイルを変更することなく、部品名(ターゲット名)やピン名の変換処理が可能です。(『6.4.3 DCF変換フィルタ』参照)

6. 4 テストパターンの生成

6. 4. 1 テストパターンの種別

テストパターンにはインフラ試験用のテストパターンとベクタ試験用のテストパターンの2種類があります。インフラ試験 とは部品の接続試験の実行前にバウンダリスキャン用のシリアルライン(TCK, TMS, TDi, TDo)の正常性を確認す るための試験です。ベクタ試験はスキャンチェインを構成するバウンダリスキャン対応デバイス間で接続される全ての ネットに対して接続試験を行います。通常はメニューの[生成]-[インフラ&テストベクタ生成]を実行することで、両方 のテストベクタを生成します。

バウンダリスキャン試験

インフラ試験:シリアルライン(TCK, TMS, TDi, TDo)の正常性を確認 ベクタ試験 :スキャンチェイン間のネットの試験

テストベクタの生成過程は以下のようになっています。これらの各プロセスはメニューの[生成]から個別に実行可能です。

[]内は[生成]メニュー内の個別生成コマンド名

Debug Solutions

1. プロジェクトファイルに設定されたDCFの読み込み

プロジェクトファイルに設定されたDCFディレクトリ内のDCFファイルをアプリケーションに読み込みます。スキャン チェインは回路図データ(EDIFデータ)とこのDCFデータを元に作成されるため、回路内のスキャンチェインを構 成する全てのDCFデータを用意する必要があります。

DCFディレクトリ内のDCFファイルは全て取り込まれます。このため不必要なDCFファイルはDCFディレクトリから 削除することにより、処理速度の向上とメモリ使用量の低下につながります。

-> 出力ファイル:Parts Define File[. ptd]

2. EDIFファイルの読み込み/解析

プロジェクトファイルに設定されたEDIFファイル名を読み取り、EDIFの文法解析処理を行います。Powerデバッガが扱うことができるEDIFには制限があります。EDIFファイルの制限についてを参照願います。

3. スキャンチェインの解析

読み取ったEDIFファイルとDCFディレクトリに格納されているDCFファイルの情報をもとにスキャンチェイン(バウ ンダリスキャンデバイスのデイジーチェーンの情報)を作成します。スキャンチェインを構成する最初の情報はプロ ジェクトファイルに設定されたInterface Port名です。

- -> 出力ファイル:Scan Chain File[. scf]
- 4. スキャンチェインセル間のネットリスト(スキャンネット)の作成

スキャンチェイン情報とEDIFファイルの情報からスキャンチェインを構成するデバイス間で接続される信号(スキャンネット)を抽出します。バウンダリスキャン試験はスキャンネットに対して実行されます。

- -> 出力ファイル:Net List File[. nlf]
- 5. インフラ試験テストベクタの生成

スキャンチェイン情報からバウンダリスキャンバス(TCK, TMS, TDi, TDo)の正常性を確認するための試験であるインフラ試験のテストベクタを生成します。

-> 出力ファイル:Infra Patern File[. ipf]

6. ベクタ試験テストベクタの生成

スキャンネットをバウンダリスキャン試験するためのテストパターンを生成します。試験の方法はALL"L"試験(オ プション)、ALL"H"試験(オプション)、SET"H"試験, SET"L"試験のテストパターンを生成します。各試験の詳

Debug Solutions

細は試験の実行を参照して下さい。

-> 出力ファイル:Patern File[. ptf]

バウンダリスキャンテストベクタデータには、PCBデバッガ内部で決まった値が割りつけられています。Oは単体試験、 1はインフラ試験、2はSET"L"試験、3はSET"H"試験に予約されています。バウンダリスキャン試験は4以降で各ネット毎に試験データが生成されます。

エラーがあるとその発生要因を表示し、以降の処理を中断します。エラー要因を特定し修正してください。エラーの詳細 はエラーメッセージを参照して下さい。

6. 4. 2 スキャンチェインデバイスのマスク処理

Power Debuggerではスキャンチェインデバイスを簡単に無効化することができます。この機能は、スキャンチェイン を構成するデバイスに未搭載部品がある状態でデバッグを進める場合に有効です。

バウンダリスキャン試験はバウンダリスキャンデバイスをカスケード接続することで試験が可能となります。このため部 品の未実装でスキャンチェインのデバイスがひとつでも欠けると試験ができなくなります。

この場合ボード上の未搭載デバイスのTDiとTDoをストラップすることで試験が可能となりますが、スキャンチェインの 構成が変わっているため、Edifを変更する必要があります。Power DebugerではEdifを変更することなくスキャンチ ェインデバイスのマスク処理が可能です。



図6.4.2-1 スキャンチェインデバイスのマスク処理

マスク処理方法

(1)通常と同じように [生成]-[インフラ&テストベクタ] を実行し、スキャンチェインを生成します。

(2)スキャンチェインビューでマスクするデバイスにマウスを持っていき、右クリックのメニューから [指定した部品の有効/無効化]を選択します(又は [編集]-[指定した部品の有効/無効化] を選択します)。



図6.4.2-2 スキャンチェインデバイスのマスク処理操作

(3)すべてのマスクする部品に対して(2)の操作を行ないます。

(4) [生成] - [インフラ&テストベクタ]を実行し、スキャンチェインを再生成します。

(5)未搭載部品のマスク状態でテストベクダが生成されます。

6. 4. 3 DCF変換フィルタ

Power DebuggerはEDIFファイルのCell NameとDCFファイルのTarget Nameを用いて、EDIFファイルとDCF ファイルの部品情報をリンクします。このため、EDIFファイルのCell Nameと対応するDCFファイルのTarget Nam eを一致させる必要があります。DCFファイルのTarget Nameを直接テキストディタを用いて編集することも可能ですが、DCF変換フィルタダイアログを用いることで、アプリケーション内部で変換することが可能となります。(アプリケーション内部のデータを変更するだけで、DCFファイルは変更しません。)

Debug Solutions

6. 4. 4 EDIF変換フィルタ

ターゲットボードの回路でバウンダリスキャン試験を動作させるために無効化する必要のある部品(デバッグのため、E DIFファイルには存在するが、ターゲットボードには未実装でスキャンチェインに影響するような部品等)を仮想的に無 効化します。本機能で無効化されたインスタンスはスキャンチェインの生成及びスキャンネットの生成で無視されます。

例えば図6.4.4-1に示すようにバウンダリスキャン回路の動作が未確認のデバイスを無効にする為、ターゲットボードの回路で、バウンダリスキャンパスをバイパスできるようにOΩの抵抗をパラレルに接続する場合があります。この様な回路のEDIFを用いた場合、テストパターン生成オプションで『2ピン部品を受動素子とする』を使用した場合、スキャンパスが一筆書きにならない為、エラーを検出します。このような場合、未実装部品であるR1の抵抗を無効化することで、正常にバウンダリスキャンパスの認識を行うことができます。



図6.4.4-1 バウンダリスキャンパスエラー検出回路例

6. 4. 5 テストパターン生成オプション

テストパターンの生成方法は以下のオプションがあります。ターゲットボードの条件によって設定する値を変更して下さい。

(1)ALL"L"/"H"試験

ALL"L"/"H"試験は試験対象ネットの全ての出力ピンから同じレベルを出力します。一つのネットに、複数の出力(入出力)ピンが存在する場合、出力信号が衝突することになります。バスコンフリクトによるエラーを避けるためには、このチェックボックスを外してテストパターンを生成して下さい。

(2)出力イネーブル付きのピンからデータを出力する。

このチェックボックスをオンにすることで、試験対象外時、出力イネーブル付きのピンに対してもデータを出力しま す。複数の出力ポートに共通の出力イネーブルで制御するデバイスを用いて、ワイヤードORを構成している場合、 出力信号が確定できないため、ベクタ試験でエラーになる可能性があります。このような回路を使用している場合 で、エラー出力を回避する場合は、このチェックボックスをオフにします。



図6.4.5-1 出力イネーブルピン制御

(3)Nopステートの幅

Nopステートの幅をクロック数で指定します。Nopステートは、バウンダリスキャン試験の各シーケンス間のNo O peration状態で、TCK, TMS, TDi, TDoの各信号は全て"H"となります。デフォルト値は "8" です。

(4)Test-Logic-Resetの幅

TAPコントローラのリセットは、TMS信号に"H"レベルを5CLK以上、保つことで実行されます。Test-Logic-R esetシーケンスのクロック数を指定します。デフォルト値は "6" です。この値はTRST信号が有効な場合、TRS T信号のリセットパルス幅になります。

(5)2ピンの部品を受動素子とする

スキャンネットの作成は、Edifで定義されている全てのネットから、スキャンチェインを構成しているデバイス間の ネットを検索します。そのため、スキャンチェインデバイス間にダンピング抵抗やロジックが含まれていると、スキャ ンネットとは認識されません。しかしダンピング抵抗などの受動素子は論理には関係せず、バウンダリスキャン試 験では無視することができます。Power Debuggerでは、スキャンネット作成時、2ピンの部品は受動素子として 扱い、直列に挿入されている2ピン部品は無視してスキャンネットを作成することができます。



図6.4.5-2 2ピン部品の受動素子

(6)ピン属性のチェック

このチェックボックスをオンにすることで、テストパターン作成時、Edifファイルで定義されているポートの属性(IN PUT | OUTPUT | INOUT)とDCFファイルで定義されているピンの属性(in | out | outz | i nout)の比較を行い、異なる場合はワーニングメッセージを出力します。

(7)BSDLファイルでSafe Bitが有効なピンも試験対象とする。

双方向バッファの方向制御ピンのような入出カピンの方向制御を行なうピンを試験対象にした場合、外部から入 カされた値により、対応する入出カピンの方向制御が変化し、バスコンフリクトを起こす可能性があります。このた め、BSDLファイルでSafe Bitが設定され、control信号でない場合は、強制的にSafe Bitを設定し、バスコンフ リクトをおこさないようにしています。しかしこのチェックボックスをオンにすることで、Safe Bitが設定され、contr ol信号でないピンに対してもバウンダリキャン対象とします。この場合、バスコンフリクトを起こし、バウンダリスキ ャン試験でエラーが発生する可能性があります。



図6.4.5-3 Safe Bitが有効なピンの試験対象の可否

6.5 試験の実行

6.5.1 試験の種類

バウンダリスキャン試験にはインフラ試験とベクタ試験があります。通常はメニューの [試験]-[インフラ&ベクタ試験] を実行することで、インフラ試験に引き続き、ベクタ試験を実行します。

☆インフラ&ベクタ試験:

インフラ試験に引き続き、ベクタ試験を実行します。インフラ試験でエラーが検出された場合でも、ベクタ試験は実 行されます。

☆インフラ試験:

バウンダリスキャン用のシリアルライン(TCK, TMS, TDi, TDo)の正常性を確認するための、インフラ試験を実行します。

☆ベクタ試験

スキャンチェイン間のネットの試験を行なうベクタ試験を実行します。既にインフラ試験がパスしている場合、この ベクタ試験のみを実行することで、試験時間の短縮が可能です。

6.5.2 インフラ試験

インフラ試験は部品の接続試験の実行前にバウンダリスキャン用のシリアルライン(TCK, TMS, TDi, TDo)の正常 性を確認するための試験です。PCBデバッガではインストラクションキャプチャコードの比較を行なうことでインフラ試 験を実現しています。インフラ試験が正常終了した場合は、スキャンチェインウィンドウのスキャンパスは緑色(デフォル ト)に変わります。又エラーを検出するとスキャンパスは赤色(デフォルト)に変化します。

Debug Solutions

6.5.3 ベクタ試験

スキャンチェインを構成するバウンダリスキャン対応デバイス間で接続される全てのネットに対して接続試験を行いま す。バウンダスキャン試験は以下のシーケンスで実行されます。

(1)ALL"L"試験:被試験対象の全てのネットの全ての出力ピンから"L"レベルを出力し、試験対象ネットの入力ピンのモニタレベルが"L"であることを確認します。
 一つのネットに対して、複数の出力(入出力)ピンが存在する場合、出力信号が衝突する為、バスコンフリクトによるエラー出力の可能性がある場合は [ツール]-[オプション] で、この機能は無効にしてください。

(2) ALL"H"試験:被試験対象の全てのネットの全ての出カピンから"H"レベルを出力し、試験対象ネットの入力ピンのモニタレベルが"H"であることを確認します。
 一つのネットに対して、複数の出力(入出力)ピンが存在する場合、出力信号が衝突する為、バスコンフリクトによるエラー出力の可能性がある場合は [ツール]-[オプション] で、この機能は無効にしてください。

- (3) SET"H"試験:被試験対象ネット以外の出力ピンには"L"を出力しておき、被試験対象ネットの出力ピンから"H" レベルを出力(複数の出力(入出力)ピンがある場合、ネット内の他の出力ピンは"L"に、入出力ピ ンはHi-Zにセットされる)し、ネット内の入力ピンのモニタレベルが"H"レベルであることを確認す ることで、試験対象ネットの正当性を調査します。
- (4)SET"L"試験:被試験対象ネット以外の出力ピンには"H"を出力しておき、被試験対象ネットの出力ピンから"L" レベルを出力(複数の出力(入出力)ピンがある場合、ネット内の他の出力ピンは"L"に、入出力ピ ンはHi-Zにセットされる)し、ネット内の入力ピンのモニタレベルが"L"レベルであることを確認す るすることで、試験対象ネットの正当性を調査します。

6.6 試験結果の解析

6. 6. 1 インフラ試験

Command View にインフラ試験の詳細結果が表示されます。

Command View	
PD > PD > Analyze Infra Test Pattern!! PD > ***********************************	0000001) 0000001) 1000001) 10000001) 10000001) 00010001

図6.6.1-1 インフラ試験結果(Command View)

インフラ試験はBSDLファイルで定義されたCapture Codeの値と、試験の結果TDoから取得した値を比較すること でバウンダリスキャン制御信号の正当性のチェックを行います。Command Viewにはスキャンチェインを構成する各 デバイスの試験結果が表示されます。

インフラ試験結果は、Scan Chain View にも表示されます。正常終了した場合、Scan Chain View のスキャン パスは緑色(デフォルト)に変わります。又エラーを検出するとスキャンパスは赤色(デフォルト)に変化します。



図6.6.1-2 インフラ試験結果(Scan Chain View)

上記例の場合、D13のデバイスとD15のデバイスの間のスキャンパスにエラーのある可能性があります。

6. 6. 2 ベクタ試験

試験結果が出力ピンと入力ピンの信号レベルで表示されます。出力信号値は "O"、"1"で示されます。又、入力ピンでの期待値が "L"、"H"、"X"で示され、実際の検出値が "O"、"1"で表示されます。



図6.6.2-1 ベクタ試験結果(Command View)

試験結果のCommand Viewへの表示は、対象となるピン名に続いて、信号値が表示されます。ピン表示は出力ピン、 Hi-Z制御付き出力ピン、入出力ピン、入力ピン及び入出力ピンの順に表示されます。

試験の解析結果は Net List View に表示されます。Status の項目にエラーの有無を表示し、エラーのある行 は赤色(デフォルト)で表示します。又"Result Analyzation"の項目に考えられるエラー要因を表示します。

🚺 Bo	Boundary Scan Net List View															
No.	Status	Parts ID	Output Parts Name	Pin Name	Pin No.	\rightarrow	Parts ID	Input Parts Name	Pin Name	Pin No.	All_L	AII_H	Set_H	Set_L	Result Analyzation	
061	OK!!					->	D4	PLD	"D6"	019	-	-	OK!	OK!	Good	
062	OK!!	D1	SN74BCT8245BA	″ A8″	015	->	D12	PLD	″ D7″	018	-	-	OK!	OK!	Good	
063	OK!!					->	D4	PLD	″D7″	018	-	-	OK!	OK!	Good	
064	OK!!	D12	PLD	″ D7″	018	->	D12	PLD	" D7"	018	-	-	OK!	OK!	Good	
065	OK!!					->	D4	PLD	" D7"	018	-	-	OK!	OK!	Good	
066	OK!!	D4	PLD	″ D7″	018	->	D12	PLD	" D7"	018	-	-	OK!	OK!	Good	
067	OK!!					->	D4	PLD	" D7"	018	-	-	OK!	OK!	Good	
068	OK!!	D5	SN74BCT8244	"1Y1"	002	->	D4	PLD	"INO"	040	-	-	OK!	OK!	Good	
069	OK!!	D5	SN74BCT8244	″1Y2″	003	->	D4	PLD	"IN1"	041	-	-	OK!	OK!	Good	
070	OK!!	D5	SN74BCT8244	″1Y3″	004	->	D4	PLD	"IN2"	042	-	-	OK!	OK!	Good	
071	OK!!	D5	SN74BCT8244	"1Y4"	005	->	D12	PLD	" INO"	040	-	-	OK!	OK!	Good	
072	OK!!	D5	SN74BCT8244	"2Y1"	007	->	D12	PLD	"IN1"	041	-	-	OK!	OK!	Good	
073	OK!!	D5	SN74BCT8244	"2Y2"	008	->	D12	PLD	"IN2"	042	-	-	OK!	OK!	Good	
074	OK!!	D11	SN74BCT8244	"2Y4"	010	->	D12	PLD	"RWN"	065	-	-	OK!	OK!	Good	
075	OK!!					->	D4	PLD	"RWN"	065	-	-	OK!	OK!	Good	
076	Error	D4	PLD	"TWT"	044	->	D12	PLD	" IN3"	043	-	-	OK!	Error	Clip Vcc or Pin Floating	
077	OK!!	D11	SN74BCT8244	"2Y2"	008	->	D15	SN74BCT8244	"1GN"	001	-	-	OK!	OK!	Good	ΞĪ.
																الق
┸															<u>•</u>	11
				_												

図6.6.2-2 ベクタ試験結果(Scan Chain View)

下表試験解析結果の表示メッセージと考えられるエラー要因を示します。尚詳細については『バウンダリスキャンの原 理 5章 エラー要因の検出』を参照して下さい。

	表示メッセージ	考えられるエラー要因
1	Good	問題なし。
2	Clip Gnd or Pin Floating	対象となるネットがGndに接地しているか、接続不
		良の可能性があります。
3	Clip Vcc or Pin Floating	対象となるネットがVccにショートしているか、接続不
		良の可能性があります。
4	Clip Gnd or Output Pin Floating	対象となるネットがGndに接地しているか、接続元
		のピンが接続不良の可能性があります。
5	Clip Vcc or Output Pin Floating	対象となるネットがVccにショートしているか、接続元
		のピンが接続不良の可能性があります。
6	Clip Gnd or Input Pin Floating	対象となるネットがGndに接地しているか、接続元
		のピンが接続不良の可能性があります。
7	Clip Vcc or Input Pin Floating	対象となるネットがVccにショートしているか、接続元
		のピンが接続不良の可能性があります。
8	Input Pin Floating(Detect Gnd)	接続先のピンが接続不良の可能性があります。
9	Input Pin Floating(Detect Vcc)	接続先のピンが接続不良の可能性があります。
10	Connect Other or Output Pin Floating	他のネットにショートしているか接続元のピンが接続
		不良の可能性があります。
11	Connect Other Pattern	スキャンネット以外のネットにショートしている可能性
		があります。
12	Connect Test Pattern	スキャンネットのネットにショートしている可能性があ
		ります。

表6.6.2-1 表示メッセージに対するエラー要因

Net List View の行をダブルクリックすると対応するネットが Scan Chain View に表示されます。この時、その ネットにエラーが発生している場合その要因を表示します。



Debug Solutions

以下にエラーメッセージに対するScan Chain Viewに表示されるエラー表示を示します。

	表示メッセージ	エラー表示
1	Good	
2	Clip Gnd or Pin Floating	Gnd? Float? Float?
3	Clip Vcc or Pin Floating	Vcc? Float? Float?
4	Clip Gnd or Output Pin Floating	Gnd? Float?
5	Clip Vcc or Output Pin Floating	Vcc?
6	Clip Gnd or Input Pin Floating	Gnd? Float?
7	Clip Vcc or Input Pin Floating	
8	Input Pin Floating(Detect Gnd)	Float?
9	Input Pin Floating(Detect Vcc)	Float?
10	Connect Other or Output Pin Floating	Connet Other Patter
11	Connect Other Pattern	Connet Other Patter
12	Connect Test Pattern	Connet Test Pattern

表6 6 2-2 表示メッセージに対するエラー表示

6. 6. 3 結果の厳密なチェック

試験結果の解析時、エラーとなったネットが他のネットにショートしている可能性があると判断した場合、他のスキャン ネットの試験結果パターンとの比較を行うことにより、ショートしている可能性のあるネットを検索します。一致したパタ ーンを検出すると、Net List View の"Result Analyzation"の項目に部品(インスタンス)IDとポート名を表示しま す。

比較範囲はスキャンネットのみです。

このオプションを選択した場合、エラー数が多いと、処理時間が長くなる場合があります。

第7章 単体試験

第7章

単体試験

7. 単体試験

7.1 単体試験について

Powerデバッガでは、バウンダリスキャン機能を用いて、スキャンチェイン上のデバイスのピン値の設定/モニタが可能です。この機能を単体試験機能と呼んでいます。

単体試験には以下の3種類がサポートされています。

・入力ピンのモニタ

スキャンチェイン上のデバイスの入力ピンのモニタを行います(出力値は変化しません)。モニタした値はBSDLビューのピン値に表示されます。入力ピンモニタはダイナミック動作(デバイスの内部論理が動作中の試験)が可能です。

・出カピン設定(データ出力後入カピンのモニタを行なう)

スキャンチェイン上のデバイスの出力ピンに設定された値を出力します。出力する値はBSDLビューのセル値で設 定を行います。入力ピンのモニタはデータ出力後、データのサンプリングを行います。モニタした値はBSDLビュー のピン値に表示されます。出力ピン設定はダイナミック動作はできません。スキャンチェインに接続されているデバ イスは全てバウンダリスキャンテスト状態になり、バウンダリスキャンリセットまで、出力ピンは設定された値を出力 し続けます。

・出カピン設定(入カピンのモニタ後データ出力を行なう)

入カピンデータのモニタ後、スキャンチェインデバイスの出カピンに、設定された値を出力します。出力する値はB SDLビューのセル値で設定を行います。モニタした値はBSDLビューのピン値に表示されます。出カピン設定はダ イナミック動作はできません。スキャンチェインに接続されているデバイスは全てバウンダリスキャンテスト状態にな り、バウンダリスキャンリセットまで、出カピンは設定された値を出力し続けます。

Debug Solutions

第7章 単体試験

7.2 BSDLビューにおける設定

BSDLビューではバウンダリスキャンデバイスの設定及びモニタを行なうことができます。以下にその方法を示します。



図7.2-1 BSDL Viewにおける設定

[ピンの状態表示]

バウンダリスキャン試験により、モニタしたピンの状態を表示します。

	表示	状態				
1	0	"L"レベルが出力あるいは入力されています。				
2	1	"H"レベルが出力あるいは入力されています。				
3	Z	ハイインピーダンス状態となっています。				
4	L	バスコンフリクトが発生しています。(入力が"L"、出力が"H"になっています。)				
5	н	バスコンフリクトが発生しています。(入力が"H"、出力が"L"になっています。)				
6	E	エラー(DCFファイルの表記にエラーがあります。)				
※出ナ	※出力ピンの表示レベルはデバイスから出力が予想される値であり、実際の値をモニタしているわ					

表7.2-1 ピンの状態表示

けではありません。

Power Debugger User's Manual Debug Solutions

[入力レジスタ]

スキャンレジスタを左クリックすることで、"-" \rightarrow "X" \rightarrow "L" \rightarrow "H" \rightarrow "-" …… とトグルします。又は 右クリックでメニュー表示されます。

- バウンダリスキャン試験で期待値との比較を行い、一致しない場合はネットリストビューにエラーを表示します。
- X :バウンダリスキャン試験で期待値との比較を行いません。エラーがあった場合でもエラー表示を行ないま せん。
- L :入力期待値として"L"レベルを想定します。"H"レベルの入力があった場合エラー表示を行います。GND に接続される入力信号の試験を行うことができます。
- H :入力期待値として"L"レベルを想定します。"H"レベルの入力があった場合エラー表示を行います。VCC に接続される入力信号の試験を行うことができます。

※入力レジスタの設定はバウンダリスキャン試験の時に有効であり、単体試験では意味をなしません。

[出力レジスタ]

スキャンレジスタを左クリックすることで、" – " → "0" → "1" → " – " ……とトグルします。又は右クリクでメニュー表示されます。

- :バウンダリ試験に基づいたレベルを出力します。
- 0 : 強制的に"L"レベルを出力します。
- 1 : 強制的に"H"レベルを出力します。

出力レジスタに設定した値はバウンダリスキャン試験時でも有効です。このため出力レジスタを設定することで、非バウンダリスキャンデバイスへの制御信号等を無効にする事が可能です。

※双方向ピンに接続されるレジスタについて

双方向セルもしくは、入力設定とコントロール設定が同時にされているセルは、入力と出力と要因を持つためレジスタ 表示は以下のようになります。

- ー :バウンダリスキャン試験で期待値との比較を行ってエラー判定を行い、又生成された値を出力します。
- -0 :入力ではバウンダリスキャン試験で期待値との比較を行い、出力では強制的に"L"レベルを出力します。
- -1 :入力ではバウンダリスキャン試験で期待値との比較を行い、出力では強制的に"H"レベルを出力します。
- X- :入力ではバウンダリスキャン試験で期待値との比較を行わず、出力では生成された値を出力します。
- X0 :入力ではバウンダリスキャン試験で期待値との比較を行わず、出力では強制的に"L"レベルを出力します。
- X1 :入力ではバウンダリスキャン試験で期待値との比較を行わず、出力では強制的に"H"レベルを出力します。
- L- :入力期待値として"L"レベルを想定し、出力では生成された値を出力します。
- LO :入力期待値として"L"レベルを想定し、出力では強制的に"L"レベルを出力します。
- L1 :入力期待値として"L"レベルを想定し、出力では強制的に"H"レベルを出力します。
- H- :入力期待値として"H"レベルを想定し、出力では生成された値を出力します。
- HO :入力期待値として"H"レベルを想定し、出力では強制的に"L"レベルを出力します。
- H1 :入力期待値として"H"レベルを想定し、出力では強制的に"L"レベルを出力します。

Debug Solutions

第7章 単体試験

7.3 入力ピンのモニタ

入力ピンのモニタでは、スキャンチェインデバイスの任意のピン値のモニタが可能です。この操作を実行すると、バウン ダリスキャンセルからデータを取り込み入出力ピンの状態を取得します。モニタしたデータはBSDLビューのピン値に 表示されます。入力ピンは外部からの入力状態を示し、出力ピンは内部ロジックから出力されている値を示していま す。

この操作でPCB上のデバイスの内部/外部論理は変化しませんが、バウンダリスキャンセルの強制設定によってデ ータを出力している場合は、バウンダリスキャンセルのリセットにより、出力ピン状態が変化する場合があります。

モニタ機能はバウンダリスキャン試験のSAMPLE/PRELOAD命令を実行することにより実現しています。



図7.3-1 入力ピンモニタ処理フロー

7. 4 出力ピン設定(データ出力後モニタ)

出力ピン設定ではスキャンチェイン上のデバイスの任意の出力ピン値の設定が可能です。この試験モードでは設定デ ータのデータを出力後、入力ピンのモニタを行います。

出力するデータはBSDLビューの出力レジスタ設定値が出力されます。"0"は出力ピンから"L"レベルを出力、"1"は 出力ピンから"H"レベルを出力します。"-"の時は現在内部論理から出力されている値を保持します。モニタしたデー タはBSDLビューのピン値に表示されます。入力ピンは外部からの入力状態を示し、出力ピンは内部ロジックからの出 力("-"の時)又はセル強制設定値("1"又は"0"の時)の値を示しています。

データ出力後モニタ機能はバウンダリスキャン試験のSAMPLE/PRELOAD及びEXTEST命令を実行することにより実現しています。



第7章 単体試験

7.5 出力ピン設定(モニタ後データ出力)

出力ピン設定ではスキャンチェイン上のデバイスの任意の出力ピン値の設定が可能です。この試験モードでは入力ピンのモニタ後、設定データのデータを出力を行います。

モニタしたデータはBSDLビューのピン値に表示されます。入力ピンは外部からの入力状態を示し、出力ピンは内部ロ ジックからの出力("-"の時)又は出力レジスタ設定値("1"又は"0"の時)の値を示しています。出力するデータはBS DLビューのセル設定値が出力されます。"0"は出力ピンから"L"レベルを出力、"1"は出力ピンから"H"レベルを出力 します。"-"の時は現在内部論理から出力されている値を保持します。

データ出力後モニタ機能はバウンダリスキャン試験のSAMPLE/PRELOAD及びEXTEST命令を実行することにより実現しています。





第8章 オンボードメモリ書き込み

第8章

オンボードメモリ書き込み

第8章 オンボードメモリ書き込み

8. オンボードメモリ書き込み

オンボードメモリ書き込みには、フラッシュメモリ及びシリアルEEPROMに対応しています。シリアルEEPROMは、I2 C、SPI、MicroWireに対応しています。

8. 1 オンボードメモリ書き込み条件

8. 1. 1 フラッシュメモリ書き込み条件

書き込み対象となるメモリは、書き込みの為に必要な信号がスキャンチェインデバイスに接続されている必要があります。



図8.1.1-1 オンボードフラッシュメモリ書き込み接続図

注意:インフラ試験パターン生成が完了していないとオンボード書き込み機能は有効となりません。

8. 1. 2 EEPROMメモリ書き込み条件

書き込み対象となるメモリは、書き込みの為に必要な信号がスキャンチェインデバイスに接続されている必要があります。



図8.1.2-1 オンボードEEPROM書き込み接続図

SPI, MicroWireデバイスは、DI(データ入力信号)、DO(データ出力信号)、CLKがそれぞれ、バウンダリスキャンデバイスに接続されている必要があります。各信号は、一つのバウンダリスキャンデバイスでも、複数のデバイスでも問題ありません。

注意:インフラ試験パターン生成が完了していないとオンボード書き込み機能は有効となりません。

第8章 オンボードメモリ書き込み

I2Cデバイスの接続構成を以下に示します。



図8.1.2-2 オンボードEEPROM書き込み接続図

I2Cデバイスは、SCL(クロック信号)、SDA(データ入出力信号)がそれぞれバウンダリスキャンデバイスに接続されている必要があります。I2Cの場合、SCL、SDA各信号は、それぞれ個別にHi-Z/入出力制御ができなければなりません。一つのバウンダリスキャンデバイスに接続する場合は、異なる入出力制御信号を持つピンに接続されていなければなりません。

注意:インフラ試験パターン生成が完了していないとオンボード書き込み機能は有効となりません。

8.2 フラッシュメモリ書き込み

8.2.1 書き込みファイルのタイプ

書き込みファイルは、以下のROMフォーマットをサポートしています。

(1)Intel-HEX

IntelーHEXファイルはバイナリデータを16進数で表現したアスキーキャラクタで構成されます。ファイルはテキストファ イルなので、汎用のテキストエディタで編集可能です。各レコードは以下のフォーマットで構成されます。

Recod	Load	Offset	RecType	Info or	Data	ChkSum
Mark ":"	RecLen					
(1 char)	(2 char)	(4 char)	(2 char)	(n -	char)	(2 char)
Recor	d Mark	: レコー	「の開始を意	意味する "∶"(コロ	レン)を記述します	•
Load	RecLen	: Info o	or Data フ	ィールドのデータバ	イト数を表します。	
Offset		: データ	の先頭アドレ	レスを示します。		
RecTy	pe	: レコー	ドタイプを表	します。		
		"00"	: データレ	コード		
		"01"	: エンドレコ	コード		
		"02"	: 拡張アド	レスレコード		
		"03"	: 開始アド	レスレコード		
		"04"	:拡張リニ	アアドレスレコード		
		"05"	: 開始リニ	アアドレスレコード		
Info o	or Data	: 1Byte	のデータを2	2ByteのASCIIで表	します。最初のデ・	-タが上位を
ChkSu	um	: Load の補数	RecLen ; をとった値。	からInfo or Data	の最後のデータ	きで加算し、そ

詳細は『Intel Hexadecimal Object File Format Specification』を参照下さい。

(2)Motorola-S

Motorola-Sファイルはバイナリデータを16進数で表現したアスキーキャラクタで構成されます。ファイルはテキストファイルなので、汎用のテキストエディタで編集可能です。各レコードは以下のフォーマットで構成されます。

Туре	Count	Address	Data	ChkSum			
(2 char)	(2 char)	(4, 6, 8 char)	(0~64 char)	(2 char)			
Туре	Type : レコードタイプを表します。 "SO": モジュール名やバージョン等のコメントが定義されます。 "S1": バイトデータの定義 "S2": 16ビットデータの定義						

	"S3" : 32ビットデータの定義
	"S5" : S1/S2/S3レコードのレコード数の定義
	"S7" : 32ビットデータのスタートアドレス定義
	"S8" : 16ビットデータのスタートアドレス定義
	"S9" : バイトデータのスタートアドレス定義
Count	: Address フィールドからレコード末尾までのデータバイト数を表します。
Address	: データの先頭アドレスを示します。
Data	: 1Byteのデータを2ByteのASCIIで表します。最初のデータが上位を示します。
ChkSum	: Count フィールドから Data フィールドの最後のデータまで加算した値。

(3)Binary

Binaryデータ形式は、アドレス情報を持たない、連続したデータが保存されているバイナリデータファイルです。

(4)Text

Text データ形式は、アドレス情報を持たない、連続したデータをテキストファイルフォーマットで表現したデータファイル です。行内にはデータを16進数文字(0~9、A, B, C, D, E, F, a, b, c, d, e, f)で記述します。データ間は(空白/ タブ)で間を空けてもあけなくてもかまいません。但し、1行内のデータ数はフラッシュメモリのデータビット幅の倍数でな ければなりません。

例:8ビットデータのフラッシュメモリ2チップ構成の場合

98ef 449d 09ac be8e ff8f 56ed	7743 ab9e
38bc 4682 ffe7 8032	4487 f4a8 320d
043A503bBBdE00433356	
E467 098c 554F	

8.2.2 デバイスファイル

デバイスファイルは、デバイスの制御情報が記述されている, Power Debugger独自フォーマットのテキストファイル です。デバイスのピン数、データの書き込みシーケンス、セクタ情報等データの書き込み/読み取りに必要な情報が記 述されおり、データの書き込み/読み取りを行うために必ず必要なファイルです。

記述の方法に関してはコマンドリファレンス付録Dを参照下さい。

デバイスファイル例:

```
Power Debugger Flash Memory Device Definition File
11
17
DevVer
                1 1
                Fujitsu
DevMaker
                MBM29F002TC
DevName
// Device definition
DevDef Adr
DevDef Data
                18
                8
DevDef
                7
        Sector
DevDef CS1
                ON
                        NEG
                                Н
 ÷
                ÷
DevDef Byte
                0FF
DevDef Reset
DevDef WP
                        NEG
                0 N
                                 н
                0FF
ProgramStart
                               // program flow definition
        WriteData 0x555 0xaa
        WriteData 0x2aa 0x55
WriteData 0x555 0xa0
        WriteData #Address #Data
        :Data_Polling
        ReadData #Address
        CompData 0x80 #Data then :OK
                                                                           データ書き込み
        CompData 0x20 0x20 else :Data_Polling
                                                                             シーケンス
        .
ReadData #Address
        CompData 0x80 #Data then :OK else :Error_Det
        : OK
        End
        :Error_Det
        Error
ProgramEnd
                                // program flow definition end
                               // chip erase flow definition
ChipEraseStart
   チップイレースシーケンスを記述
                               // chip erase flow definition end
ChipEraseEnd
                                // sector erase flow definition
SectorEraseStart
 セクタイレースシーケンスを記述
SectorEraseEnd
                              // sector erase flow definition end
11
// Sector Area Range
11
                     Start_Adr
                                       End_Adr
11
                Num
                                                       Erase_Adr
//-----
                0x00000000
                                 0x0000ffff
                                                 0x00000000
                                                                          // 64KByte
Sector 0
                0x00010000
                                 0x0001ffff
                                                 0x00010000
                                                                          // 64Kbyte
Sector 1
                0x00020000
                                 0x0002ffff
                                                 0x00020000
                                                                          // 64KByte
Sector
        2
Sector
        3
                0x00030000
                                 0x00037fff
                                                 0x00030000
                                                                          // 32KByte
                                                                          // 8KByte
// 8KByte
Sector
        4
                0x00038000
                                 0x00039fff
                                                 0x00038000
Sector
        5
                0x0003a000
                                 0x0003bfff
                                                 0x0003a000
                0x0003c000
                                                                          // 16KByte
                                 0x0003ffff
                                                 0x0003c000
Sector 6
```

8.2.2-1 デバイスファイル例

Debug Solutions

8.2.3 チップ構成

オンボード書き込みで書き込み可能なフラッシュメモリのアドレス信号は、10~32ビット、データ信号は8/16/32ビ ットの組み合わせとなります。又チップ数は1/2/4チップに対応しています。組み合わせ可能なチップ構成を以下に 示します。

No	アドレス	データ	チップ容量	チップ数	全体容量
1				1 Chip	1 KByte
2		8 bit	1 KByte	2 Chip	2 KByte
3	10 64			4 Chip	4 KByte
4		16 54		1 Chip	2 KByte
5				2 Chip	4 KByte
6		32 bit	4 KByte	1 Chip	4 KByte
7				1 Chip	2 KByte
8	- 11 bit -	8 bit	2 KByte	2 Chip	4 KByte
9				4 Chip	8 KByte
10		10 54		1 Chip	4 KByte
11			4 KByte	2 Chip	8 KByte
12		32 bit	8 KByte	1 Chip	8 KByte
13	12 bit			1 Chip	4 KByte
14		8 bit	4 KByte	2 Chip	8 KByte
15				4 Chip	16 KByte
16		IZ DI			1 Chip
17			8 KByte	2 Chip	16 KByte
18		32 bit	16 KByte	1 Chip	16 KByte
19				1 Chip	8 KByte
20		8 bit	8 KByte	2 Chip	16 KByte
21	10 64			4 Chip	32 KByte
22		16 hit	16 KBite	1 Chip	16 KByte
23			TO REVIE	2 Chip	32 KByte
24		32 bit	32 KByte	1 Chip	32 KByte
25				1 Chip	16 KByte
26		8 bit	16 KByte	2 Chip	32 KByte
27	14 64			4 Chip	64 KByte
28		16 54	20 KD++	1 Chip	32 KByte
29		IO DIL	32 KByte	2 Chip	64 KByte
30	1	32 bit	64 KByte	1 Chip	64 KByte

No	アドレス	データ	チップ容量	チップ数	全体容量
31				1 Chip	32 KByte
32		8 bit	32 KByte	2 Chip	64 KByte
33	15 bit			4 Chip	128 KByte
34	15 bit	16 hit	64 K Puta	1 Chip	64 KByte
35		10 01	04 NByle	2 Chip	128 KByte
36		32 bit	128 KByte	1 Chip	128 KByte
37				1 Chip	64 KByte
38	-	8 bit	64 KByte	2 Chip	128 KByte
39	16 hit			4 Chip	256 KByte
40		10 54	100 // 0.1	1 Chip	128 KByte
41	-		128 KByte	2 Chip	256 KByte
42		32 bit	256 KByte	1 Chip	256 KByte
43	17 bit			1 Chip	128 KByte
44		8 bit	128 KByte	2 Chip	256 KByte
45				4 Chip	512 KByte
46		10 10		1 Chip	256 KByte
47			256 KByte	2 Chip	512 KByte
48		32 bit	512 KByte	1 Chip	512 KByte
49				1 Chip	256 KByte
50		8 bit	256 KByte	2 Chip	512 KByte
51	10 64			4 Chip	1 MByte
52		16 bit	512 KButa	1 Chip	512 KByte
53			512 KByle	2 Chip	1 MByte
54		32 bit	1 MByte	1 Chip	1 MByte
55				1 Chip	512 KByte
56		8 bit	512 KByte	2 Chip	1 MByte
57	10 hit			4 Chip	2 MByte
58	19 01	16 bit		1 Chip	1 MByte
59			T MByte	2 Chip	2 MByte
60		32 bit	2 MByte	1 Chip	2 MByte
61				1 Chip	1 MByte
62		8 bit	1 MByte	2 Chip	2 MByte
63	20 hit			4 Chip	4 MByte
64		16 54	2 MB:+-	1 Chip	2 MByte
65		זוס טונ		2 Chip	4 MByte
66		32 bit	4 MByte	1 Chip	4 MByte

第8章 オンボードメモリ書き込み

No	アドレス	データ	チップ容量	チップ数	全体容量
67		8 bit		1 Chip	2 MByte
68			2 MByte	2 Chip	4 MByte
69	21 bit			4 Chip	8 MByte
70		16 bit		1 Chip	4 MByte
71			4 MByte	2 Chip	8 MByte
72		32 bit	8 MByte	1 Chip	8 MByte
73	22 bit	8 bit	4 MByte	1 Chip	4 MByte
74				2 Chip	8 MByte
75				4 Chip	16 MByte
76		16 bit		1 Chip	8 MByte
77			8 WByte	2 Chip	16 MByte
78		32 bit	16 MByte	1 Chip	16 MByte
79			8 MByte	1 Chip	8 MByte
80	23 bit	8 bit		2 Chip	16 MByte
81				4 Chip	8 MByte
82		16 54	16 MByte	1 Chip	16 MByte
83				2 Chip	32 MByte
84		32 bit	32 MByte	1 Chip	32 MByte
85	24 bit	8 bit	16 MByte	1 Chip	16 MByte
86				2 Chip	32 MByte
87				4 Chip	64 MByte
88		16 bit		1 Chip	32 MByte
89			32 MByte	2 Chip	64 MByte
90		32 bit	64 MByte	1 Chip	64 MByte
91	25 bit	8 bit	32 MByte	1 Chip	32 MByte
92				2 Chip	64 MByte
93				4 Chip	128 MByte
94		16 54		1 Chip	64 MByte
95			04 MByte	2 Chip	128 MByte
96		32 bit	128 MByte	1 Chip	128 MByte
97	26 bit	8 bit	64 MByte	1 Chip	64 MByte
98				2 Chip	128 MByte
99				4 Chip	256 MByte
100		16 -		1 Chip	128 MByte
101				2 Chip	256 MByte
102		32 bit	256 MByte	1 Chip	256 MByte

Debug Solutions

Power Debugger User's Manual

64

No	アドレス	データ	チップ容量	チップ数	全体容量
103		8 bit	128 MByte	1 Chip	128 MByte
104	27 bit			2 Chip	256 MByte
105				4 Chip	512 MByte
106		16 bit	OFC MD.t.	1 Chip	256 MByte
107			200 MIDyle	2 Chip	512 MByte
108		32 bit	512 MByte	1 Chip	512 MByte
109		8 bit	256 MByte	1 Chip	256 MByte
110	28 bit			2 Chip	512 MByte
111				4 Chip	1 GByte
112		16 bit	512 MByte	1 Chip	512 MByte
113				2 Chip	1 GByte
114		32 bit	1 GByte	1 Chip	1 GByte
115			512 MByte	1 Chip	512 MByte
116		8 bit		2 Chip	1 GByte
117	- 29 bit			4 Chip	2 GByte
118		10.11	1 GByte	1 Chip	1 GByte
119		16 bit		2 Chip	2 GByte
120		32 bit	2 GByte	1 Chip	2 GByte
121			1 GByte	1 Chip	1 GByte
122		8 bit		2 Chip	2 GByte
123				4 Chip	4 GByte
124		16 54	2 GByte	1 Chip	2 GByte
125				2 Chip	4 GByte
126		32 bit	4 GByte	1 Chip	4 GByte
127	31 bit	8 bit	2 GByte	1 Chip	2 GByte
128				2 Chip	4 GByte
129				4 Chip	8 GByte
130			4 GByte	1 Chip	4 GByte
131				2 Chip	8 GByte
132		32 bit	8 GByte	1 Chip	8 GByte
133	- 32 bit	8 bit	4 GByte	1 Chip	4 GByte
134				2 Chip	8 GByte
135				4 Chip	16 GByte
136		10.11	0.05 ·	1 Chip	8 GByte
137		16 bit	8 GByte	2 Chip	16 GByte
138		32 bit	16 GByte	1 Chip	16 GByte

第8章 オンボードメモリ書き込み

8.2.4 接続信号

フラッシュメモリのオンボード書き込みに必要な接続信号を以下に示します。

Address00~xx	:	アドレス信号
Data00~xx	:	データ信号。8/16/32ビットのバス幅に対応しています。
Chip Select1(CS1)	:	チップセレクト信号
Chip Select2(CS2)*1	:	チップセレクト信号(複数のCS信号をもつデバイスの場合)
Write Enable1(WE1)	:	書き込みイネーブル信号
Write Enable2(WE2)*1	:	書き込みイネーブル(複数のWE信号をもつデバイスの場合)
Output Enable1(OE1)	:	読み出しイネーブル信号
Output Enable2(OE2)*1	:	読み出しイネーブル信号(複数のOE信号をもつデバイスの場合)
Byte ^{*1}	:	BYTE/WORD切り替え信号
Reset ^{*1}	:	リセット信号
Write Protect(WP)*1	:	書き込み禁止信号

*1 : オプション信号です。

8.2.5 メモリデータ書き込み

フラッシュメモリ書き込みのダイアログは、[書き込み設定], [ポート設定], [メモリ読み出し], [オプション]の4つのタ ブに分かれています。以下にその設定内容の詳細を示します。

[書き込み設定] :	書き込みデバイス名	、データファイルの指定なと	ビ基本的な設定を行ないます。
------------	-----------	---------------	----------------

- : 書き込みデバイスが接続されたバウンダリスキャンデバイスのポートを指定します。 [ポート設定]
- [メモリ読み出し] : フラッシュメモリに書き込まれたデータの読み出しを行ないます。
- [オプション] : フラッシュメモリ書き込みのオプション設定を行ないます。

Power Debugger User's Manual Debug Solutions
(1)書き込み設定

ash Memory 書き込み	
書き込み設定 ボートの設定 メモリ読み出し オブション	
Device File名: MBM29LV160T-Fujitsu _]
-データファイルタイプ C Intel-HEX C Motorola-S C Binary © Text	
書き込みFile名: E¥Jtag¥FlashProj¥Flash.txt File	1
 オフセットアドレス: [0x0000]	_
 ✓ アドレス範囲を有効とする 先頭アドレス: 0x0000 ※了アドレス: 0x000f 	
バイトオーダー ・ ビッグエンティアン[RISC系] C リトルエンディアン[x86系]	
□ データ書き込み後、ベリファイを行う。	
●動作 ○ セクタ消去&プログラム ● プログラムのみ ○ チップ消去	7
	_
	2007.0
ower Debugger終了 キャンセル 書き込みへ	17

図8.2.5-1 書き込み設定画面

①DeiceFile名:

書き込みを行なうフラッシュメモリのデバイスファイルを指定します。デバイスファイルは、デバイスの制御情報が 記述されているテキストファイルです。最新のデバイスファイルは弊社ホームページからダウンロードするか、担 当窓口にお問い合わせください。

②データファイルタイプ:

フラッシュメモリに書き込むデータのファイルタイプを指定します。書き込みデータファイルは、Intel-HEX 、Mot orola-S 、Binary , Text データフォーマットから選択します。 各フォーマットの詳細は、『8.2.1 書き込み ファイルのタイプ』を参照ください。

③書き込みファイル名:

フラッシュメモリに書き込むデータのファイル名を指定します。「②データファイルタイプ」で指定されたフォーマット に基づいた読み込み処理がされます。 ④オフセットアドレス:

書き込むデータのオフセットを指定します。「③書き込みファイル名」で指定したデータファイルから読み取ったデータに対するアドレスに、ここで指定したオフセットアドレスを加算したアドレスにデータを書き込みます。Binary形式や、Text形式のデータフォーマットのようにアドレス情報を持たないデータの書き込みをする場合に指定します。

⑤アドレス範囲:

書き込みファイル名で指定したデータファイルから読み取ったデータの全てではなく、一部分のみの書き込みを行 なう場合、「アドレス範囲を有効とする」のチェックボックスをオンにして、範囲指定を行ないます。「④オフセットアド レス」で指定した値を加算したアドレスに対して範囲の評価を行ないます。

⑥バイトオーダー:

16ビット以上の複数バイト幅の書き込みを行なう際のバイトオーダーを指定します。ビッグエンディアンでは、最上位のバイト(MSB)から順番にアドレスの昇順(0→1→2→3→....)に格納するような書き込みを行ない、リトルエンディアンでは、最小位のバイト(LSB)から順番にアドレスの昇順(0→1→2→3→....)に格納するような書き込みを行ないます。

32bitデータ構成の場合, ビッグエンディアンでは、データファイルの書き込みデータ4バイト(Ox12 - Ox34 - Ox56 - Ox78) は

bit31-bit24 : 0x12 bit23-bit16 : 0x34 bit15-bit08 : 0x56 bit07-bit00 : 0x78

の順に格納されます。

32bitデータ構成の場合, リトルエンディアンでは、データファイルの書き込みデータ4バイト(0x12 - 0x34 - 0x56 - 0x78) は

bit31-bit24 : 0x78 bit23-bit16 : 0x56 bit15-bit08 : 0x34 bit07-bit00 : 0x12

の順に格納されます。

一般的にRISC系CPUでは、ビッグエンディアンが、Intel系CPUでは、リトルエンディアンが採用されています。

⑦データ書き込み後のベリファイ:

このチェックボックスをオンにすると、書き込みワード単位に読み出しを行ない、書き込みの確認を行います。

⑧動作:	
セクタ消去&プログラム	: 書き込み対象とセクタを消去した後、データの書き込みを行ないます。
プログラムのみ	: データの書き込みを行ないます。あらかじめ、書き込みセクタの消去がされている
	必要があります。
チップ消去	: チップ全体の消去を行ないます。

(2)ポートの設定

		1000 a.C.		
ートの設定 ンスタンスID: IC1		▼ ポート	ID: A13	•
Flash Memory - Addres	s ->	インスタンス名	ポート名	
Adr00	->	IC1	A11	
Adrui Adrui	->		A12 A19	
Adr03	->	IC1	A14	
Adr04	->	IC1	A15	•
Flash Memory - Data	->	インスタンス名	ポート名	▲
Data00	->	IC2	A11	
DataUI DataUI	2	102	A12 012	
Data03	->	IC2	A14	
Data04	->	IC2	A15	-
		5		
Flash Memory - Control	->	インスタンス名	ポート名	A
Chip1-CS1	->	IC1	B21	
Chip1-082	-2	Not Use	Not Use	
Chip1-OE1	K	Not Llog 1	Not Llee	
Unipi Ocz	/	NULUSE	Not Use	1

図8.2.5-2 ポート設定画面

①構成アドレスビット幅:

ターゲットボード上で使用している書き込み対象のフラッシュメモリのアドレス信号のビット幅を指定します。デバイスのアドレス線の数ではありません。例えば、8ビットデータ幅で256KByteの容量をもつデバイス(アドレス信号数:18本)のアドレス信号16本をバウンダリスキャンデバイスに接続して、64KByteのみを使用している場合は、アドレスビット幅:16bitを指定します。

②構成データビット幅:

ターゲットボードで使用している書き込み対象のフラッシュメモリのデータ信号のビット幅とチップ構成を指定します。

 8 Bit x 1 Chip
 (8Bit データバス幅)

 8 Bit x 2 Chip
 (16Bit データバス幅)

 8 Bit x 4 Chip
 (32Bit データバス幅)

 16 Bit x 1 Chip
 (16Bit データバス幅)

 16 Bit x 2 Chip
 (32Bit データバス幅)

 32 Bit x 1 Chip
 (32Bit データバス幅)

 32 Bit x 1 Chip
 (32Bit データバス幅)

の中から選択します。

③インスタンス ID/ポートID:

設定するインスタンスIDを選択します。

設定するインスタンスIDとポートIDを選択し、割りつけたいAddress信号又はData信号又はControl信号をクリックしてハイライト表示した後、 [選択した項目の変更] ボタンをクリックすると対象の信号が設定されます。

設定した項目の設定を解除する場合は、設定解除したいAddress信号又はData信号又はControl信号をクリックしてハイライト表示した後、 [選択した項目の削除] ボタンをクリックすると対象の信号が設定解除されます。

④From File. . . ボタン/To File. . . ボタン:

ポート設定情報の読み出し/保存を行います。DPFファイルは、フラッシュメモリのポート設定情報を記述したテキストファイルです。[From File...]ボタンで保存されたポート設定情報を読み出します。[To File...]ボタンで設定されたポート設定情報をDPFファイルに保存します。

DPFファイルは、[To File...]ボタンでテンプレートを作成して、テキストエディタを用いて、必要な箇所を変更 することで、間単に作成することができます。生成されたDPFファイルは、他のプロジェクトでも読み込み可能です ので、共通的な回路であれば、DPFファイルをコピーして読み出すことにより、流用が可能です。 (3)メモリ読み出し

F	lash Memory 書き込み	×
	書き込み設定 ポートの設定 メモリ読み出し オプション	
	開始アドレス: 🛛 x0000 🚍 終了アドレス: 🗇 x0027 🚍	
	Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +f Ascii 00000000 : ff f	
	, メモリデータ読み取り処理中(100 %)	
ł	Power Debugger 終了 キャンセル 読み込み ヘルプ	

図8.2.5-3 メモリ読み出し画面

①開始アドレス/終了アドレス:

フラッシュメモリの読み出し開始アドレスと終了アドレスを指定します。

②読み込み:

読み込みボタンをクリックすると、フラッシュメモリからデータの読み込みを行ないデータの表示を行ないます。読 み込み実行中は、[中止]に表示が変わります。読み込み実行中に、[中止]ボタンをクリックすると、処理が中止 されます。

(4)オプション

Flash Memory 書き込み	x
書き込み設定 ボートの設定 メモリ読み出し オプション	
□ ロマンドビューにステータス出力を行う。 □ エラーのポップアップ出力を行う。	
_Intel-HEX Format File	
▼ Sumチェックを行う。 ▼ 規定外のレコードタイプをエラーとする。	
Motorola-S Format File レージングを行う。 レージングのシートになくづきてラーンする。	
 ✓ S5レコードのチェックを行う。 	
プログラム実行時の無限ループ保護時間: 10 🕂 秒	
チップ消去時の無限ループ保護時間: 30 34	
Power Debugger 終了 キャンセル 書き込み ヘルブ	

図8.2.5-4 オプション設定画面

①コマンドビューにステータス出力を行なう:

フラッシュメモリのアクセスシーケンスログを表示します。正常に書き込みが出来ない時等、デバッグ時に使用しま す。

②エラーのポップアップ出力を行なう:

フラッシュメモリのアクセス時、エラーが発生すると、ポップアップ表示を行ない、処理を一時停止します。

③IntelーHEX フォーマットファイル オプション:

Sumチェックを行なう。	:	各L		·ド毎	にチョ	⊏ック	サム	を計算し、レコード最後尾のChkSum値
		との	比較	を行	ないる	ます。		
規定外のレコードタイプをエラーとする。	:	00	01	02	03	04	05	以外のレコードタイプを検出すると、エラ
		ーを	出力	しまっ	す。			

詳細は、『8.2.1 書き込みファイルのタイプ』を参照ください。

Power Debugger User's Manual Debug Solutions

④Motorola-Sフォーマットファイル オプション:

Sumチェックを行なう。	: 各レコード毎にチェックサムを計算し、レコード最後尾のChkSum値
	との比較を行ないます。
規定外のレコードタイプをエラーとする。	: S0 S1 S2 S3 S4 S5 S7 S8 S9 以外のレコードタイプを
	検出すると、エラーを出力します。
S5レコードのチェックを行なう。	: S5レコードはS1/S2/S3レコードのレコード数の定義を行ないま
	す。S1/S2/S3レコードの数をカウントし、S5レコードの値と比較
	を行ないます。

詳細は、『8.2.1 書き込みファイルのタイプ』を参照ください。

⑤プログラム実行時の無限ループ保護時間:

フラッシュメモリプログラム実行中、プログラムシーケンスの無限ループに陥らないように、プログラムシーケンスの実行時間を制限します。この制限時間を秒で指定します。

⑥チップ消去時の無限ループ保護時間:

フラッシュメモリチップ消去実行中、プログラムシーケンスの無限ループに陥らないように、プログラムシーケンスの実行時間を制限します。通常、チップ消去はセクタ書き込みやセクタ消去に比較して時間が必要なため、別に 指定するようになっています。

8.2.6 書き込み時間の見積もり

フラッシュメモリの書き込み時間は以下の式で示されます。

 Tes
 :セクタ消去時間

 Ns
 :消去セクタ数

 Twr
 :データ書き込み時間

 Nd
 :書き込みデータ数

書き込み時間[Tall] = Tes × Ns + Twr ×Nd

ここで、セクタ消去時間[Tes]は

セクタ消去時間[Tes] = セクタ消去アクション時間[Tea] + セクタ消去確認時間[Ter]

となり、セクタ消去確認時間[Ter]はデバイスのセクタ消去時間で、デバイスに依存し、通常、数10msec~数秒程度 です。セクタ消去アクション時間[Tea]はデバイス定義ファイルで定義されるセクタ消去シーケンスのアクセス回数に依 存します。

仮に

WriteData 0x555 0xaa WriteData 0x2aa 0x55 WriteData 0x555 0x80 WriteData 0x555 0xaa WriteData 0x2aa 0x55 WriteData 0x555 0x10

で示されるセクタ消去シーケンスの場合、6回の書き込みシーケンスが必要です。この場合、

セクタ消去アクション時間[Tea] = データ出力時間[Tout] × 6

となります。

データ出力時間[Tout]はバウンダリスキャンを用いた一連のデータ出力処理時間で以下の式で表されます。

データ出力時間[Tout] = スキャンチェインレジスタ数 × (1/書き込みCLK周波数[Fclk])
 × アクセスシーケンス数(2) + CPU処理時間

となります。スキャンチェインレジスタ数は、スキャンチェインを構成するバウンダリスキャンレジスタの数です。フラッシュメモリに接続されないデバイスはバイパスモードとなるので、フラッシュメモリに接続されるデバイスのバウンダリスキャンレジスタ数の合計値となります。(強制出力やピンモニタ機能が有効になっているデバイスは、バイパスモードとな

らず合計値の対象となります。)

書き込みCLK周波数[Fclk]は書き込みアダプタから出力されるTCKの周波数で、[ツール]-[アダプタの設定]で測定できます。アクセスシーケンス数は書き込み及び読み取り処理で必要なシーケンス数で2となります。

又、CPU処理時間はホストPCのCPU処理能力/OSによって異なってきます。

データ書き込み時間[Twr]はデバイス定義ファイルで定義されるセクタデータ書き込みシーケンスのアクセス回数に依存します。

仮に

WriteData 0x555 0xaa WriteData 0x2aa 0x55 WriteData 0x555 0xa0 WriteData #Address #Data :Data_Polling ReadData #Address

で示される書き込みシーケンスの場合、4回の書き込みと1回の読み取りが必要です。この場合、

データ書き込み時間[Twr] = データ出力時間[Tout] × (4 + 1)

となります。データ出力時間[Tout]は、セクタ消去アクション時間[Tea]で説明した値です。

仮に、消去/書き込みシーケンスのアクセス回数は上記の値として、以下の条件の場合 CPU : セレロン 400MHz 書き込み周波数 : 50KHz スキャンチェインセル数: 880 書き込みセクタ数 : 1 セクタ当りのデータ数 : 8KByte メモリデータ幅 : 16bit セクタ消去時間 : 1sec : 10ms (CPU処理時間は、CPUの種類、動作周波数、OS等の条件で異なります。) CPU処理時間 = 880 × 1/50KHz × 2 + 10ms データ出力時間[Tout] = 45.2 msとなり、書き込み時間[Tall]は 書き込み時間[Tall] = 45.2ms × 6 + 1sec + 45.2ms × 5 × (8K / 2) = 927 sec ≒ 15分 となります。

8.2.7 書き込み時間の高速化

バウンダリスキャンを用いたフラッシュメモリの書き込みは、スキャンチェイン接続されたバウンダリスキャンセルに対し てシリアルで設定するために、非常に低速度になってしまいます。しかし、フラッシュメモリに接続するデバイスの選択 や、スキャンチェインデバイスの設定を変えることで高速化することができます。以下の例で示します。



図8.2.7-1 フラッシュメモリの接続

(1)フラッシュメモリに接続される、バウンダリスキャンデバイスの数を少なくする。

フラッシュメモリ書き込み処理を実行時、フラシュメモリに接続されないスキャンチェイン上のバウンダリスキャンデバイ スは、バイパス設定され、1ビットのシフトレジスタになります。フラッシュメモリに接続される、バウンダリスキャンデバイ スの数を少なくすることで、シフトレジスタの数を少なくすることができ、高速化につながります。

(2) 強制設定やモニタ設定を解除する。

フラッシュメモリ書き込み処理を実行時、フラシュメモリに接続されないスキャンチェイン上のバウンダリスキャンデバイ スは、バイパス設定されますが、信号の強制出力設定や、モニタ設定があると、バイパス設定されません。フラッシュメ モリアクセス時は、不要な強制設定やモニタ設定を解除してください。

8.3 シリアルEEPROM書き込み

8.3.1 対応デバイス

対応しているシリアルEEPROMの種別は以下の通りです。

- (1) I2C
- (2) SPI
- (3) Microwire

	I2C	SPI	Microwire
必要な信号	SCL, SDA	SCK, CS, SI, SO	SK, CS, SI, SO
オプション信号	Protect	Protect Hold	Protect

8.3.2 書き込みファイルのタイプ

書き込みファイルは、以下のROMフォーマットをサポートしています。

- (1)Intel-HEX
- (2)Motorola-S
- (3)Binary
- (4)Text

詳細は、『8.2.1 書き込みファイルのタイプ』を参照ください。

8.3.3 デバイスファイル

デバイスファイルは、デバイスの制御情報が記述されている、Power Debugger独自フォーマットのテキストファイル です。デバイスの種別、データ容量、インストラクションセットなどの書き込み/読み取りに必要な情報が記述されおり、 シリアルEEPROMデータの書き込み/読み取りを行うために必ず必要なファイルです。

記述の方法に関してはコマンドリファレンス付録Eを参照下さい。

8.3.4 接続信号

EEPROMのオンボード書き込みに必要な接続信号を以下に示します。

(1)I2C

SCL	: クロック信号。双方向信号です。 必須指定です。
SCD	: データ信号。双方向信号です。 必須指定です。
Protect	: ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。 <i>オプションです。</i>

(2)SPI

SCK	:	クロック信号。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
CS	:	データ信号。双方向信号です。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
SI	:	データ信号。PC->ターゲットデバイス方向の信号です。 <i>必須指定です。</i>
SO	:	データ信号。ターゲットデバイス->PC方向の信号です。 <i>必須指定です。</i>
Protect	:	ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。 <i>オプションです。</i>
Hold	:	ホールド信号PC->ターゲットデバイス方向の信号です。 <i>オプションです。</i>

(3) Microwire

SK	: クロック信号。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
CS	: データ信号。双方向信号です。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
SI	: データ信号。PC->ターゲットデバイス方向の信号です。 <i>必須指定です。</i>
SO	: データ信号。ターゲットデバイス->PC方向の信号です。 <i>必須指定です。</i>
Protect	: ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。 オプションです。

8.3.5 メモリデータ書き込み

EEPROMメモリ書き込みのダイアログは、[書き込み設定], [ポート設定], [メモリ読み出し], [オプション]の4つの タブに分かれています。以下にその設定内容の詳細を示します。

[書き込み設定]	: 書き込みデバイス名、データファイルの指定など基本的な設定を行ないます。
[ポート設定]	: 書き込みデバイスが接続されたバウンダリスキャンデバイスのポートを指定します。
[メモリ読み出し]	: EEPROMメモリに書き込まれたデータの読み出しを行ないます。
[デバイス設定]	: EEPROMメモリデバイスのデバイス定義設定を行ないます。
[オプション]	: EEPROMメモリアクセスのオプション設定を行ないます。

(1)書き込み	設定
---------	----

PROM 書き込み
書き込み設定 ポート定義 読み出し デバイス設定 オブション
▼ データファイルを指定する
データファイルタイプ C Intel-HEX C Motrola-S C Binary © Text
書き込みFile名: E¥Jtag¥FlashProj¥Flash.txt File

(16)進数表記)
オフセットアドレス : 0x0000 📑
┌ ▼ アドレス範囲を有効とする
先頭アドレス: 0x0000 王 終了アドレス: 0x00ff 王
ハイトオーター ● ビッグエンディアン[RISC系] ● リトルエンディアン[x86系]
□
 ● チップ消去&プログラム ● プログラムのみ ● チップ消去
書き込み禁止コマンド送信 書き込み禁止解除コマンド送信
wer Debugger 終了 キャンセル 書き込み ヘルプ

図8.3.5-1 書き込み設定画面

①データファイルを指定する:

書き込みを行なうデータのファイルを指定します。このチェックボックスをオフにすると、書き込みデータ エディット ボックスに入力したデータが有効となります。

②データファイルタイプ:

EEPROMメモリに書き込むデータのファイルタイプを指定します。書き込みデータファイルは、Intel-HEX 、M otorola-S 、Binary , Text データフォーマットから選択します。 各フォーマットの詳細は、『8.2.1 書き込 みファイルのタイプ』を参照ください。

③書き込みファイル名:

EEPROMメモリに書き込むデータのファイル名を指定します。「②データファイルタイプ」で指定されたフォーマット に基づいた読み込み処理がされます。

④書き込みデータ:

EEPROMメモリに書き込むデータを指定します。「①データファイルを指定する」チェックボックスをオフにするとで 有効になります。「⑤オフセットアドレス」で指定したアドレスを先頭としたアドレスで順番に書き込みが行われます。 データは、16進数表記(先頭のOxは必要ありません)の空白で区切られた、8桁以内の O~9、A、B、C、D、E、 F(a、b、c、d、e、f)の文字のみ認識可能となっています。又1文字入力は2桁として、3文字入力は4桁として、5 ~7文字入力は、8桁として認識されます。

⑤オフセットアドレス:

書き込むデータのオフセットを指定します。「③書き込みファイル名」で指定したデータファイルから読み取ったデ ータに対するアドレスに、ここで指定したオフセットアドレスを加算したアドレスにデータを書き込みます。Binary形 式や、Text形式のデータフォーマットのようにアドレス情報を持たないデータの書き込みをする場合に指定しま す。

⑥アドレス範囲:

書き込みファイル名で指定したデータファイルから読み取ったデータの全てではなく、一部分のみの書き込みを行 なう場合、ここで範囲指定を行ないます。「④オフセットアドレス」で指定した値を加算したアドレスに対して範囲の 評価を行ないます。

⑦バイトオーダー:

16ビット以上の複数バイト幅の書き込みを行なう際のバイトオーダーを指定します。ビッグエンディアンでは、最 上位のバイト(MSB)から順番にアドレスの昇順(0→1→2→3→....)に格納するような書き込みを行ない、リ トルエンディアンでは、最小位のバイト(LSB)から順番にアドレスの昇順(0→1→2→3→....)に格納するよう な書き込みを行ないます。

```
32bitデータ構成の場合, ビッグエンディアンでは、データファイルの書き込みデータ4バイト(0x12 - 0x34 - 0x56 - 0x78) は
```

```
bit31-bit24 : 0x12
bit23-bit16 : 0x34
bit15-bit08 : 0x56
bit07-bit00 : 0x78
の順に格納されます。
32bitデータ構成の場合, リトルエンディアンでは、データファイルの書き込みデータ4バイト(0x12 - 0x34
- 0x56 - 0x78) は
bit31-bit24 : 0x78
bit23-bit16 : 0x56
bit15-bit08 : 0x34
bit07-bit00 : 0x12
の順に格納されます。一般的にRISC系CPUでは、ビッグエンディアンが、Intel系CPUでは、リトルエンディアン
```

⑧データ書き込み後のベリファイ:

このチェックボックスをオンにすると、書き込みワード単位(ページモードの時はページ単位)に読み出しを行ない、 書き込みの確認を行います。

⑨動作:

チップ消去&プログラム	チップ全体の消去を実行した後、データの書き込みを行ない	ます。
プログラムのみ	対象範囲のデータの書き込みを行ないます。	
チップ消去	チップ全体の消去を行ないます。	

⑨書き込み禁止コマンド送信:

SPI, MicroWire デバイスのWriteEnable コマンドを送信します。I2Cデバイスでは、無効となります。

⑩書き込み禁止解除コマンド送信:

SPI, MicroWire デバイスのWriteDisable コマンドを送信します。。I2Cデバイスでは、無効となります。

(2)ポートの設定

EEPROM 書き込み			×			
書き込み設定 ポート定義 読みと	出し デバイス設定 オブション	1				
	インスタンスID:	ボートID:				
SCK (Master -> EEPROM) :	IC2	A21				
CS (Master -> EEPROM) :	IC2	A15	-			
SI (Master -> EEPROM) :	IC1 💌	A12	•			
SO (Master <- EEPROM):	IC1 💌	B11	•			
Protect (Master -> EEPROM) :	IC2	A19	•			
Hold (Master -> EEPROM) :	IC2	A11	-			
- Chip Select信号プロパティー ▼ CS信号負論理						
Protect信号プロパティ ✓ Protect信号がスキャンラ ✓ Protect信号負論理	テェインに接続されている。					
Hold信号ブロバティ ▼ Hold信号がスキャンチェインに接続されている。 ▼ Hold信号負論理						
Power Debugger 終了	キャンセル	書き込み	ヘルプ			

図8.3.5-2 ポート設定画面

Power Debugger User's Manual

①インスタンス ID/ポート ID:

EEPROMメモリデバイスの各信号が接続される、バウンダリスキャンチェインデバイスのインスタンス ID とポート ID を指定します。接続されるEEPROMメモリデバイスの種別により、対応した信号名が表示されます(EEPRO Mメモリデバイスの種別は、『デバイス設定』タブで指定します)。表示される信号名は、以下のようになります。

(1)I2C

SCL	:	クロック信号。双方向信号です。 <i>必須指定です。</i>
SCD	:	データ信号。双方向信号です。 必須指定です。
Protect	:	ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。オプションです。

(2)SPI

SCK	:	クロック信号。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
CS	:	データ信号。双方向信号です。PC->ターゲットデバイスへの信号です。必須指定です。
SI	:	データ信号。PC->ターゲットデバイス方向の信号です。 <i>必須指定です。</i>
SO	:	データ信号。ターゲットデバイス->PC方向の信号です。 <i>必須指定です。</i>
Protect	:	ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。オプションです。
Hold	:	ホールド信号PC->ターゲットデバイス方向の信号です。 <i>オプションです。</i>

(3) Microwire

SK	:	クロック信号。PC->ターゲットデバイスへの信号です。 <i>必須指定です。</i>
CS	:	データ信号。双方向信号です。PC->ターゲットデバイスへの信号です。必須指定です。
SI	:	データ信号。PC->ターゲットデバイス方向の信号です。 <i>必須指定です。</i>
SO	:	データ信号。ターゲットデバイス->PC方向の信号です。 <i>必須指定です。</i>
Protect	:	ライトプトレクト信号。PC->ターゲットデバイス方向の信号です。 <i>オプションです。</i>

☆ChipSelect信号プロパティ:

②CS信号負論理

チップセレクト信号をもつデバイス(SPI、Microwire)のCS信号線の論理を設定します。チップセレクト信号が 負論理の場合このチェックボックスをオンにします。

☆Protect信号プロパティ:

③Protect信号がスキャンチェインに接続されている。

ターゲットデバイスのProtect信号がスキャンチェインに接続されている場合このチェックボックスをオンにします。データの書き込み実行時、Protect信号を制御して、書き込み可能状態とします。

④Protect信号負論理

プトテクト信号線の論理を設定します。Protect信号が負論理の場合このチェックボックスをオンにします。

☆Hold信号プロパティ:

⑤Hold信号がスキャンチェインに接続されている。

ターゲットデバイス(SPI)のHold信号がスキャンチェインに接続されている場合このチェックボックスをオンにし

ます。データの書き込み実行時、Hold信号を制御して、書き込み可能状態とします。

⑥Hold信号負論理

ホールド信号線の論理を設定します。Hold信号が負論理の場合このチェックボックスをオンにします。

(3)メモリ読み出し

'ROM 書き込a まき込み設定	, ポー	小定事		売み	出L	,] :	デバ	ተአ	設定	E	オブ	ýs)	21					ļ
開始アドレス	: 0)×000()			-	÷		終	77	'ドレ	יג:	0	×00	028		- i	
Address	+0 -	+1 +2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	Ascii		
00000000 : 00000010 : 00000020 :	ff f ff f ff f	ff ff ff ff ff ff	ff ff ff	ff ff ff	ff ff ff	ff ff ff	ff ff ff	ff ff ff	ff ff									
メモリデータ	読み	取归文	⊥∃里□	₽ (`	100	%)												
wer Debug	aer			終	7		1	*	- tr).	1也	1	1		読み	応入	* 1	AJI.	,

図8.3.5-3 メモリ読み出し画面

①開始アドレス/終了アドレス:

EEPROMメモリの読み出し開始アドレスと終了アドレスを指定します。

②読み込み:

読み込みボタンをクリックすると、EEPROMメモリからデータの読み込みを行ないデータの表示を行ないます。読み込み実行中は、[中止]に表示が変わります。読み込み実行中に、[中止]ボタンをクリックすると、処理が中止されます。

(4)デバイス設定

IOM 書き込み		
き込み設定 ポート定義	読み出し デバイス設定 オ	วังรุ่ม
デバイス名: 25AA040 ベンダ名: Microchi) P	[デバイスファイルの読み取り]
○ 120(2ワイヤシリアル) 構成 bit幅: 8	フード数: 64	✓ Protect信号あり
SlaveAddress: 1010aaa	AddressByte: 1	Y
ページサイズ: 8	Target Code: 0x00	Write Time: 10 💼 msec
• SPI		
構成 bit幅: 8	▼ ワード数: 512	▼ Protect信号あり ▼ HOLD信号あり
ページサイズ: 16	AddressByte: 1	Write Time: 5 - msec
- インストラクションセット5 Write: 0000a010 Read: 0000a011	E義 Write SR: 00000001 Read SR: 00000101	WriteEn: 000000110 WriteDS: 00000100
⊂ Microwire(2□√tz≩41	70)	
構成 bit幅: 16	マリード数: 64	Protect信号あり 速結時ないい、対応
ページサイズ: 8	▼ Write Time: 10	■ msec
- - インストラクションセット5		
Write: 01aaaaaa	WriteALL: 0001xxxx	WriteEn: 0011xxxx
Read: 10aaaaaa	EraseALL: XXXXXXXX	WriteDS: 0000xxxx
Erase: xxxxxxxxx	Page Write: 11aaaaaa	

図8.3.5-4 デバイス設定画面

①デバイスファイルの読み取り:

書き込みを行なうEEPROMメモリのデバイスファイルを指定します。デバイスファイルは、デバイスの制御情報が 記述されているテキストファイルです。読み取られたパラーメータは、以下のパラメータに反映されます。 最新のデバイスファイルは弊社ホームページからダウンロードするか、担当窓口にお問い合わせください。。

②構成 bit幅(I2C, SPI, Microwire):ワードデータのビット幅を指定します。

③ワード数(I2C, SPI, Microwire): ワード数を指定します。

④Slaveアドレス(I2C):

I2Cデバイスのヘッダデータ(デバイスコードとスレーブアドレス)を指定します。その文字には、"01aAxX*"が 使用されます。又文字数は、7Bitアドレスフォーマットの時は7文字、10ビットアドレスフォーマットのときは10文

字で構成されます。

Slave Address 定義 :

- 0 : デバイスコードの"L"レベルを設定します。
- 1 : デバイスコードの"H"レベルを設定します。
- a : スレーブアドレス(アドレスピンと比較)
- A : "a"と同じ
- x : 無視
- X : x と同じ
- * : 拡張アドレス設定

⑤アドレスバイト数(I2C, SPI):

ワードアドレスのバイト数を指定します。

6ページサイズ (I2C, SPI, Microwire):

EEPROMデバイスが、ページアクセスをサポートしている場合、ページサイズをバイト数で指定します。ページア クセスをサポートしていない場合は、データワード数のバイト数を設定します(8bitデータであれば1、16bitデータ であれば2を設定)。

⑦Target Code(I2C):

アクセスする、ターゲットデバイスのスレーブアドレスを設定します。SlaveAddressの"A"(又は"a")で定義され たビットにここで設定された値が出力されます。

8Write Time(I2C, SPI, Microwire):

EEPROMデバイスの書き込み時間をmsec単位で設定します。EEPROMの書き込みは、データ書き込み後、 次のデータを書き込むまで一定時間、間隔を空ける必要があります。PowerDebuggerがデバイスにアクセスす る際、アクセスとアクセスの間に何もしない状態を出力します(NOP処理と呼んでいます)。通常、このNOP処理 でソフトウェア処理でWriteTimeを満足しますが、デバイスのWriteTime が短い場合や、高速なアクセスポート を使用した場合、NOPによる書き込み待ち合わせ時間がWriteTime[msec]より短くなる場合があります。この 場合処理を、継続すると、正常に書き込みが行われない場合があります。この場合、オプションタブの『I2Cオプ ション書き込み確認実行』をチェックするか、オプションタブの『NOPステートの幅』の設定を大きくしてください。

⑨Protect信号あり(I2C, SPI, Microwire):

ターゲットデバイスが WriteProtect 機能を制御するための信号を持っている場合、このチェックボックスを有効 とします。スキャンチェインデバイスに接続されているかどうかは、『ポートの設定』タブ「Protect信号プロパティ」 で設定します。

10Hold信号あり(SPI):

ターゲットデバイスが Hold 機能を制御するための信号を持っている場合、このチェックボックスを有効とします。 スキャンチェインデバイスに接続されているかどうかは、『ポートの設定』タブ「Hold信号プロパティ」で設定しま す。

①連続読み出し対応(Microwire):

ターゲットデバイス(Microwire)がデバイス内部のアドレスオートインクリメント機能を持っており、連続読み出しが可能な場合、このチェックボックスをオンにします。

12インスラクションセット定義

Write	:	書き込みインストラクションセットを指定します。 必須指定です。
Read	:	読み出しインストラクションセットを指定します。 <i>必須指定です。</i>
Erase	:	データ消去インストラクションセットを指定します。 <i>オプションです。</i>
WriteSR	:	ステータスレジスタ書き込みインストラクションセットを指定します。 <i>オプションです。</i>
ReadSR	:	ステータスレジスタ読み出しインストラクションセットを指定します。 <i>オプションです。</i>
WriteALL	:	チップ全体同一データ書き込みインストラクションセットを指定します。オプションです。
EraseALL	:	チップ全体データ消去インストラクションセットを指定します。 <i>オプションです。</i>
PageWrite	:	ページ書き込みインストラクションセットを指定します。 <i>オプションです。</i>
WriteEn	:	書き込み許可インストラクションセットを指定します。 <i>オプションです。</i>
WriteDs	:	書き込み禁止インストラクションセットを指定します。 <i>オプションです。</i>

※インスラクションセットが未定義で指定しない場合、Parameter2 に "x"を指定してください。

(5)オプション

EEPROM 書き込み	×
書き込み設定 ポート定義 読み出し デバイス設定 オプション	
□ コマンドビューにステータス出力を行う。 □ エラーのボッブアップ出力を行う。 □ Intel-HEX Format File	
☑ Sumチェックを行う。 ☑ 規定外のレコードタイプをエラーとする。	
Motorola-S Format File ✓ Sumチェックを行う。 ✓ S5レコードのチェックを行う。	
書さ込み確認保護時間: 2 <u>-</u> 秒	
NOPステートの幅: 2 一 CLK	
120オプション ▼ ページ書き込みモード ▼ 連続読み出しモード ▼ 書込確認実行	
SPはプション ステータスレジスタ語み出し ステータスレジスタ書き込み ⁽⁾	
 ✓ 書き込み前に自動的に書き込み可能状態とする。 ✓ ページ書き込みモード ✓ 書き込み後に自動的に書き込み禁止状態とする。 ✓ 連続読み出しモード 	
MicroWireオブション 「書き込み前に自動的に書き込み可能状態とする。 「「ページ書き込みモード 「書き込み後に自動的に書き込み禁止状態とする。 「「連続読み出しモード 「読み取り時もProtect信号を解除する。	
- Power Debugger 終了 キャンセル 書き込み ヘルブ	

図8.3.5-5 オプション設定画面

①コマンドビューにステータス出力を行なう:

フラッシュメモリのアクセスシーケンスログを表示します。正常に書き込みが出来ない時等、デバッグ時に使用します。

②エラーのポップアップ出力を行なう:

フラッシュメモリのアクセス時、エラーが発生すると、ポップアップ表示を行ない、処理を一時停止します。

③Intel-HEX フォーマットファイル オプション:

Sumチェックを行なう。 : 各レコード毎にチェックサムを計算し、レコード最後尾のChkSum値 との比較を行ないます。 規定外のレコードタイプをエラーとする。: 00 01 02 03 04 05 以外のレコードタイプを検出すると、エラ ーを出力します。

Power Debugger User's Manual

詳細は、『8.2.1 書き込みファイルのタイプ』を参照ください。

④Motorola-Sフォーマットファイル オプション:

Sumチェックを行なう。	:	各レコード毎にチェックサムを計算し、レコード最後尾のChkSum値
		との比較を行ないます。
規定外のレコードタイプをエラーとする。	:	S0 S1 S2 S3 S4 S5 S7 S8 S9 以外のレコードタイプを
		検出すると、エラーを出力します。
S5レコードのチェックを行なう。	:	S5レコードはS1/S2/S3レコードのレコード数の定義を行ないま
		す。S1/S2/S3レコードの数をカウントし、S5レコードの値と比較
		を行ないます。

詳細は、『8.2.1 書き込みファイルのタイプ』を参照ください。

⑤書き込み確認保護時間:

EEPROMデバイス(I2C, Microwire)プログラム実行時、データの書き込み確認を実行する際のタイムアウト時間を設定します。I2C デバイスプでは、Ackの応答確認を、Microwireデバイスでは、SDO信号からの書き込み 完了フラグの確認を行います。SPIデバイスは、書き込み完了を示すフラグを持たないため、デバイス設定の「Wr ite Time」で定義されている時間の待ち合わせを行う必要があります。書き込み完了の待ち合わせ時間は、⑥ 「NOPステートの幅」で指定します。

⑥NOPステートの幅:

NOPステートとは、EEPROMデバイスへのアクセスの前後に挿入される、無効データをさします。SPIデバイス は、書き込み完了を示すフラグを持たないため、デバイス設定の「Write Time」で定義されている時間の待ち合 わせを行う必要があります。待ち合わせ時間の調整はNOPステートの幅を設定することにより、書き込みアクセ スの間隔を調整することにより実現します。



図8.3.5-6 NOPステート

NOPステートによるアクセス間隔がWriteTimeより短くなる場合、以下のような警告メッセージを出力します。 「System Message!! NOPによる書き込み待ち合わせ時間が riteTime[msec] り短い可能性があります。

処理を継続しますか?」

「はい」を選択すると、正常に書き込みができない可能性があります。

☆I2Cオプション

⑦ページ書き込みモード:

このチェックボックスをオンにすると、ページ書き込みに対応しているデバイスでは、データ書き込み時、ページ書 き込みを実行します。このチェックボックスがオフの場合は、ワード単位での書き込みを実行します。

⑧連続読み出しモード:

このチェックボックスをオンにすると、データ読み出し時、連続読み出しモードでデータの読み出しを行います。このチェックボックスがオフの場合は、ワード単位にデータの読み出しを行います。

⑨書き込み確認実行:

I2Cデバイスのデータ書き込み時、データ書き込みアクセス後、Ackの応答確認を実行することで、データの書き 込み確認を行います。このチェックボックスをオフにすると、Ackの応答確認のアクセスが省略されるため、書き込 み速度が向上しますが、⑥「NOPステートの幅」を調整してアクセス間隔 WriteTime 以上になるようにしなけ ればなりません。

☆SPIオプション

⑩ステータスレジスタ読み出し:

ステータスレジスタのデータ読み出し処理を実行し、読み出したデータのポップアップ表示を行います。

①ステータスレジスタ書き込み:

設定された書き込みデータのステータスレジスタへの書き込み処理を実行します。

①書き込み前に自動的に書き込み可能状態とする:

このチェックボックスをオンにすると、書き込み処理を実行する前に自動的に書き込み許可コマンドを実行し、デバイスを書き込み可能状態にした後、データの書き込みを行います。書き込み禁止状態にあるデバイスでも、データの書き込みが実行されるので、注意が必要です。

③書き込み後に自動的に書き込み禁止状態とする:

データの書き込み処理後、書き込み禁止コマンドを実行して、デバイスを書き込み禁止状態にします。

(4)ページ書き込みモード:

このチェックボックスをオンにすると、ページ書き込みに対応しているデバイスでは、データ書き込み時、ページ書き込みを実行します。このチェックボックスがオフの場合は、ワード単位での書き込みを実行します。

15連続読み出しモード:

このチェックボックスをオンにすると、データ読み出し時、連続読み出しモードでデータの読み出しを行います。このチェックボックスがオフの場合は、ワード単位にデータの読み出しを行います。

☆Microwire オプション

16書き込み前に自動的に書き込み可能状態とする:

このチェックボックスをオンにすると、書き込み処理を実行する前に自動的に書き込み許可コマンドを実行し、デバイスを書き込み可能状態にした後、データの書き込みを行います。書き込み禁止状態にあるデバイスでも、データの書き込みが実行されるので、注意が必要です。

①書き込み後に自動的に書き込み禁止状態とする:

データの書き込み処理後、書き込み禁止コマンドを実行して、デバイスを書き込み禁止状態にします。

18読み取り時もProtect信号を解除する:

ー部のMicrowireデバイスには、データ読み取り時、Protect信号を解除する必要があるものがあります。このようなデバイスの場合、このチェックボックスをオンにします。

19ページ書き込みモード:

このチェックボックスをオンにすると、ページ書き込みに対応しているデバイスでは、データ書き込み時、ページ書き込みを実行します。このチェックボックスがオフの場合は、ワード単位での書き込みを実行します。

20連続読み出しモード:

このチェックボックスをオンにすると、データ読み出し時、連続読み出しモードでデータの読み出しを行います。このチェックボックスがオフの場合は、ワード単位にデータの読み出しを行います。

第9章 その他

第9章

その他

Power Debugger User's Manual

第9章 その他

9. その他

9.1 EDIFファイルの階層表示

プロジェクトに設定されたEDIFファイルをツリー表示します。

第9章 その他

9.2 認証コードの登録

Powerデバッガを動作させるには認証コードの入力が必要です。動作させるPCのインストールディスクのボリューム番号或いはLANカードのMACアドレスをサポートデスクに連絡下さい。折り返し認証コードを返送いたします。この認証 コードを[ヘルプ] - [シリアル番号]のSerial Numberダイアログの認証コードに入力することで使用可能となります。

☆CD-ROMでインストールする場合。



図9.2-1 認証コードの登録フロー(CD-ROM)

☆インターネットからダウンロードした場合



図9.2-2 認証コードの登録フロー(インターネット)

デバッグソリューションズがここで提供する情報は、正確かつ信頼できるものと考えておりますが、その使用に関す る責務は一切負いません。ここに記載される情報は、2005年3月におけるものです。訂正、変更、改版に追従して いない場合があります。最終的な確認はヘルプデスクにお問い合わせ下さい。

> Web http://www.debsol.com E-Mail mail@debsol.com

デバッグソリューションズ Debug Solutions