

PARTNER ユーザーズ・マニュアル
VRシリーズ
『VR4120-TP個別編』

■ PARTNERのバージョンアップ

- 最新のPARTNER-VR4120-TPは、以下のサイトよりダウンロードできます。

<http://www.midas.co.jp/products/download/program/partner.htm>

■ ご注意

- PARTNER-VR4120-TP(プログラム及びマニュアル)に関する著作権は株式会社マイダス・ラボ及び京都マイクロコンピュータ株式会社が所有します。
- 本プログラム及びマニュアルは著作権法で保護されており、弊社の文書による許可が無い限り複製、転載、改変等できません。
- お客様に設定される使用权は、PARTNER-VR4120-TP用をお客様が登録する1台のシステムにおいてのみ使用する権利を意味します。
- 本製品は、万全の注意を持って作製されていますが、ご利用になった結果については、販売会社、株式会社マイダス・ラボ及び京都マイクロコンピュータ株式会社は一切の責任を負いかねますのでご了承ください。
- 本プログラム及びマニュアルに記載されている事柄は、予告なく変更されることがあります。
- 本プログラムの変更により、このマニュアルに記載している画面のイメージと実際の画面が異なる場合があります。

■ 商標について

- MS-Windows、Windows、MS、MS-DOSは米国マイクロソフト・コーポレーションの商標です。
- そのほか本書で取り上げるプログラム名、システム名、CPU名などは、一般に各メーカーの商標です。

改訂履歴

日付 Y/M/D	Rev	内容
2000.11.14	1.00	正式初版
2001.05.20	1.01	download site 情報の変更
2002.01.10	1.10	RTE-2000-TP対応の追加 エミュレーションROMダミー、ROMコメントの変更 ROM1 ~ ROM4コメント追加
2002.11.20	1.11	以下の項を追加 6.1.3 MULTI2000での作成 1.1 PARTNER-VR4120-TPの環境設定-RTEオプション
2003.05.16	1.12	フラッシュメモリの対応デバイスにIntelタイプを追加 フラッシュメモリの環境設定に『領域設定書式 2 (FLASH_MEM2)』を追加

空白ページ

目次

はじめに.....	1
1 起動.....	3
1.1 PARTNER-VR4120-TPの環境設定.....	4
1.2 初期設定コマンド.....	7
1.3 起動時のエラーメッセージ.....	9
2 メインウィンドウ補足.....	12
2.1 メニュー.....	13
2.1.1 実行メニュー.....	13
3 チャイルドウィンドウ.....	15
3.1 レジスタウィンドウ.....	16
3.1.1 レジスタウィンドウのショートカットキー.....	17
3.1.2 レジスタウィンドウのローカルメニュー.....	17
3.1.3 レジスタウィンドウでのマウス操作.....	18
3.2 ブレークウィンドウ.....	19
3.2.1 ブレークウィンドウのショートカットキー.....	19
3.2.2 ブレークウィンドウのローカルメニュー.....	20
3.2.3 ブレークウィンドウでのマウス操作.....	20
3.3 I/Oウィンドウ.....	21
3.3.1 I/Oウィンドウのショートカットキー.....	21
3.3.2 I/Oウィンドウのローカルメニュー.....	22
3.3.3 I/Oウィンドウでのマウス操作.....	22

4 ダイアログコマンド..... 23

4.1 ダイアログボックス.....	24
4.1.1 ハードウェアブレークポイントの設定ダイアログボックス	24
4.1.2 CPU環境設定ダイアログボックス	25
4.1.3 TLBの設定ダイアログボックス.....	25
4.1.4 エミュレーションROM設定ダイアログ.....	26
4.1.5 キャッシュフラッシュダイアログボックス.....	27
4.1.6 各種状態の設定ダイアログボックス.....	27

5 コマンドリファレンス..... 29

5.1 コマンド解説の規約.....	30
5.2 機能別コマンド.....	31
5.2.1 ハードウェアの初期化	31
5.2.2 CPUリセット	31
5.2.3 環境設定コマンド.....	31
5.2.4 プログラムロード.....	32
5.2.5 ファイルリード/ライト	32
5.2.6 プログラム実行	33
5.2.7 ブレークポイント.....	33
5.2.8 ハードウェアブレークポイント.....	33
5.2.9 コード表示.....	34
5.2.10 レジスタ表示/変更.....	34
5.2.11 システムレジスタ表示/変更	34
5.2.12 メモリ表示/変更	35
5.2.13 I/Oポート入力/出力	35
5.2.14 シンボル表示/設定.....	35
5.2.15 アセンブル.....	35

5.2.16	バックトレース	35
5.2.17	キャッシュフラッシュ	36
5.2.18	Cのデータ参照/変更	36
5.2.19	文字列の定義(Cの式評価)	36
5.2.20	システム制御	36
5.2.21	システムコール	37
5.2.22	式の表示	37
5.2.23	基数変更	37
5.2.24	ロギング/バッチ	37
5.2.25	オプション設定	38
5.2.26	マクロコマンド	38
5.2.27	フラッシュメモリ管理	39

6 高級言語レベルデバッグ	41
----------------------------	-----------

6.1 GreenHills C.....	42
6.1.1 コンパイルとリンク方法.....	42
6.1.2 使用例.....	42
6.1.3 MULTI2000での作成	43
6.2 GNU C(exeGCC)	45
6.2.1 コンパイルとリンク方法.....	45
6.2.2 使用例.....	45
6.3 CodeWarrior C.....	47
6.3.1 コンパイルとリンク方法.....	47
6.3.2 使用例.....	48

7 フラッシュメモリのサポート	51
------------------------------	-----------

7.1 対応デバイス.....	52
-----------------	----

7.2 環境設定	54
7.3 書き込み動作	58
7.4 管理コマンド(ZF)	60

付録	61
-----------------	-----------

レジスタ変数	61
エラーメッセージ	62

はじめに

このマニュアルは、『PARTNER-VR4120-TP』に依存した部分の操作方法について説明しています。

その他の操作方法は『PARTNER ユーザーズ・マニュアルVRシリーズ共通編』で説明されていますので合わせて参照してください。

必要なハードウェア

- インターフェースキット

パーソナルコンピュータとRTE-1000-TPまたはRTE-2000-TPを接続するためのインターフェースキットが必要です。インターフェースキットには、『PC Cardインターフェースキット』、『PC98 Desk top PC用インターフェースキット』、『DOS/V Desk Top PC用インターフェースキット』、『PCIバス用インターフェースキット』、『10Base-TのLAN-BOX』等があります。

必要なセットアップ

- RTE for Win32

RTE for Win32のインストールマニュアルを参照してDLLのインストールを行い、使用するICEの種類と接続ポートを設定し、ハードウェアの接続確認を実施してください。

空白ページ

1 起動

PARTNER-VR4120-TPを起動するには、PARTNER-VR4120-TPの環境を設定する必要があります。

この章では、PARTNER-VR4120-TPの起動に関するオプションの設定方法を説明します。

1.1 PARTNER-VR4120-TPの環境設定

起動オプションの設定



PARTNER-VR4120-TPの起動オプションを指定します。起動オプションでは、動作モードやメモリの使用状態を変更します。(図 1)

オプション文字列

図 1 [起動オプション]ダイアログボックス

オプション文字列の個所には、ダイアログで入力した内容が文字列で表示されます。PARTNER-VR4120-TPの起動オプションは通常、RPTSETUPの[起動オプション]ダイアログボックスで指定します。指定された起動オプションはプロジェクトファイル(RPTVR4120T.KPJ)に書き込まれます。PARTNER-VR4120-TPの起動オプションを以下に示します。

-Bオプション

デバッグ情報およびマクロ領域のサイズ指定を行います。デバッグ情報量にはグローバル、ローカルシンボル、行番号情報、関数/変数の属性情報などの多くの情報が格納されます。

-B[サイズD],[サイズM]

サイズ D デバッグ情報を登録する領域のサイズ(1K バイト単位)
省略時には 1024K バイトの領域を確保します。

サイズ M マクロ定義情報を保存する領域のサイズ(1K バイト単位)
省略時には 32K バイトの領域を確保します。

-Dオプション

PARTNER-VR4120-TPのカレントディレクトリを指定します。カレントディレクトリにはコンフィギュレーションファイル(RPTVR4120T.CFG)、プロジェクトファイル(RPTVR4120T.KPJ)が存在する必要があります。

-Dディレクトリ

ディレクトリがPARTNER-VR4120-TP起動時のカレントディレクトリになります。このオプションが省略された場合は[プロパティ]ダイアログボックスの[作業フォルダ]に指定されているディレクトリ、[コマンドライン]に指定されている実行プログラムの存在するディレクトリの順にディレクトリが指定されます。

-Eオプション

コードウインドウで表示/参照するソースファイルの拡張子のデフォルトを追加指定します。

-E拡張子

拡張子がソースファイルのデフォルトの拡張子となります。拡張子にはピリオドを除く拡張子部分を指定します。このオプションが省略された場合はfile.Cまたはfile.ASMがデフォルトの指定になります。拡張子指定はセミコロン(;)で区切ることにより行えます。

-SDオプション

コードウインドウに参照するソースファイルのあるディレクトリを指定します。ディレクトリ指定はセミコロン(;)で区切ることにより複数指定できます。

-SDディレクトリ

ディレクトリがソースファイルのあるディレクトリになります。このオプションが省略された場合は-Dオプションで指定したディレクトリになります。

-TABオプション

コードウインドウのソース表示をするときのタブのサイズを指定します。エディタでタブサイズを変更したファイルを表示するのに便利な機能です。

-TABタブサイズ

タブサイズで指定される数にタブを合わせます。このオプションを省略した場合のタブサイズは8に設定されます。

-Xオプション

アセンブラまたは各種Cのデバッグモードの指定をします。PARTNER-VR4120-TPでは多くの処理系をサポートしているために各処理系を明確にする必要があります。この指定を正しく行っていない場合には、正常なデバッグができないかある

いは特定の機能が使用できないことがあります。各処理系の使用方法は、『6 高級言語レベルデバッグ』、『各処理系のマニュアル』を参照してください。

-X

アセンブラのデバッグモード(デフォルト)の指定です。このモードでPARTNER-VR4120-TPを起動するとC関連(スタックのバックトレースやローカル変数などに関連した)コマンドは利用できません。

-XC

Green Hills のデバッグモード指定をします。デバッグ情報は.OUTファイルから読み込みます。

-XG

exeGCC のデバッグモード指定をします。デバッグ情報は.OUTファイルから読み込みます。

-XCW

Code Warrior Cのデバッグモード指定をします。デバッグ情報は.OUTファイルから読み込みます。

-RTEオプション

使用するRTEのチャンネル番号を指定します。

-RTEチャンネル番号

チャンネル番号には0 ~ 3を指定します。RTEのチャンネル機能を使用しない場合は0を指定してください。

@オプション

-Dオプション以外のオプションを@で指定したファイルから読み込むことが可能です。

【例】

コマンドライン

```
C:¥RPTVR4120¥BIN¥RPTVR4120T -DC:¥SAMPLE @TEST.INF
```

ファイル内容(TEST.INF)

```
-B1024 -SDC:¥SAMPLE¥SOURCE -XC
```

1.2初期設定コマンド

コンフィギュレーションファイル(RPTVR4120T.CFG)の内容以外で、RTE-VR4120-TPの動作環境を予め設定しておく必要がある場合は、次のコマンドで初期化することができます。

各コマンドについては、コマンドリファレンス(29頁参照)、または、ハードウェア・ユーザズ・マニュアルを参照してください。

初期化で使用するコマンド

INIT	PARTNER-VR4120-TPおよびICEの初期化をします。
ENV	各種入力信号のマスクなどのCPUの各種環境を設定します。
ROM	エミュレーションROMを設定します(RTE-1000-TP用)。
ROM1 ~ ROM4	エミュレーションROMを設定します(RTE-2000-TP用)。
NC	非メモリキャッシュの領域を設定します。
NCD	非メモリキャッシュの領域を解除します。
NROM	エミュレーションメモリ割付除外領域を設定します。
NROMD	エミュレーションメモリ割付除外領域を解除します。

これらのコマンドはコマンドウィンドウで入力しますが、PARTNER-VR4120-TPの起動時に自動実行されるINIT.MCRファイルに記述しておくくと便利です。

INIT.MCRでの初期化例

INIT.MCRファイルに、以下の例のような初期化コマンドを記述しておくことで、起動毎の初期化を自動で行うことができます。

```
init
    PARTNER-VR4120-TP、ICEを初期化します。
env !nmi
    NMI信号をマスクする、に設定します。
rom BFC00000 80000 4m rom16 bus16
    エミュレーションするROMを16ビットROM、16ビットバス、4Mビット
    サイズに、そしてエミュレーションアドレス範囲をBFC00000H ~
    BFC7FFFFH番地に設定します。
nc 20000 1000
    20000番地から1000バイトの領域をメモリキャッシュの除外エリアに
    指定します。
L sample.out
    デバッグを行うプログラム「sample.exe」を読み込みます。
```

ROMコマンドはRTE-1000-TPの例です。

1.3 起動時のエラーメッセージ

PARTNER-VR4120-TPが起動時にエラーを検出した際に表示するメッセージを示します。エラー要因を取り除いてから再起動してください。



RTEの電源が入っていないか、ハードウェアの不良が考えられます。電源、接続ケーブル等を確認してください。その後、ChkRTEプログラムを実行してください。



他のアプリケーションがRTEを使用しています。または、異常終了によりRTEのコントロールプログラムは終了していません。その場合にはWindowsを再起動してください。



初期化中のRTEから応答がありません。ターゲットシステムが接続されている場合、プロセッサが動作できない可能性があります。障害を取り除いてから起動してください。



ホストカードまたはPCカードがインストールされていない可能性があります。正しくインストールされていることを確認しChkRTEプログラムを実行してください。



RTEが接続されていないか、RTEの電源が入っていないことが考えられます。電源、接続ケーブルなどを確認してください。その後、ChkRTEプログラムを実行してください。



システムの初期化ができません。RTEに電源が入っていない可能性があります。電源の接続を確認し、ChkRTEプログラムを実行してください。



何らかの異常終了によりRTEのコントロールプログラムは終了していません。Windowsを再起動するか、RTEのコントロールプログラムを強制終了してください。



RTEインストールに不備があり必要なDLLが見つかりません。RTEのSetUp CD-ROMで再インストールしてください。その後、必ずChkRTEプログラムを実行してください。

2 メインウインドウ補足

この章では、共通編と一部異なる個所について説明します。

2.1メニュー

2.1.1 実行メニュー

プログラム実行(G)	F5
カム(C)	F7
トレース(T)	F8
ステップ(S)	F10
リターン実行(U)	Ctrl+F5
ブレークポイント(P)	F9
強制ブレーク(B)	ESC
キャッシュフラッシュ(F)	
エミュレーションROM設定(M)	
CPU環境設定(V)	

[実行]メニューにはユーザプログラムの実行 / 停止に関するコマンドが収められています。

メニュー項目	機能
プログラム実行(G)	ユーザプログラムを現在のPCから実行します
カム(C)	ユーザプログラムを現在のPCからカーソルのある行まで実行します。
トレース(T)	ソース行単位、機械語単位のトレース実行をします。
ステップ(S)	ソース行単位、機械語単位のステップ実行をします。
リターン実行(U)	現在の関数の実行を終了(リターン)して呼び出し関数に戻るまで実行します。
ブレークポイント(P)	ブレークポイントの設定 / 解除をします。
強制ブレーク(B)	強制的にユーザプログラムを中断し、PARTNER-VR4120-TPに制御を戻します。(コマンド/マクロの連続実行中は連続実行を中断しません)
キャッシュフラッシュ(F)	キャッシュのフラッシュを行います。(『4.1.5 キャッシュフラッシュダイアログボックス』27頁参照)
エミュレーションROM設定(M)	エミュレーションROM領域を設定します。(『4.1.4 エミュレーションROM設定ダイアログ』26頁参照)
CPU環境設定(V)	CPUの環境値を設定します。(『4.1.2 CPU環境設定ダイアログボックス』25頁参照)

空白ページ

3 チャイルドウィンドウ

この章では、PARTNER-VR4120-TPの機能に依存したチャイルドウィンドウについて、表示や操作方法を説明します。

3.1 レジスタウインドウ

レジスタウインドウでは、CPUレジスタ、CP0レジスタ、TLBレジスタを表示します(図2)。

レジスタ名	値	表示形式
R0	0000000000000000	(0.000000e+00)
R1	000000000000FE01	(1.379828e-309)
R2(v0)	0000000000000000	(0.000000e+00)
R3(v1)	FFFFFFFFA001214E	(-1.597028e-154)
R4(a0)	000000000000FE01	(1.379828e-309)
R5(a1)	0000000000000000	(0.000000e+00)
R6(a2)	FFFFFFFFA0012178	(-1.597087e-154)
R7(a3)	0000000000000000	(0.000000e+00)
R8(t0)	FFFFFFFFA0012188	(-1.597110e-154)
R9(t1)	FFFFFFFFA0001EF9	(-1.502949e-154)
COUNT(9)	3228F811	
ENTRYHI(10)	C00000FFFFFF8000	
COMPARE(11)	FFFBFBFE	
STATUS(12)	F5408006	C:1111 FR:1 BEV:1 TS:0 SR:0 CH:0 CE:0 IM:10000000 KX:0 SX:0 UX:0 KSU:0 ERL:0
CAUSE(13)	00000010	BD:0 CE:0 IP:000000 SW:00 EXC:4(AdEL)
EPC(14)	FFFFFFFFBFC00000	
PRID(15)	00005423	
CONFIG(16)	3330E6F1	
TLB 0	M:00000000 HI:A0000000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 1	M:00000000 HI:A0002000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 2	M:00000000 HI:A0004000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 3	M:00000000 HI:A0006000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 4	M:00000000 HI:A0008000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 5	M:00000000 HI:A000A000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 6	M:00000000 HI:A000C000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 7	M:00000000 HI:A000E000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 8	M:00000000 HI:A0010000(ASID:00) LO:00000008(00000000) L1:00000000	
TLB 9	M:00000000 HI:A0012000(ASID:00) LO:00000008(00000000) L1:00000000	

図 2 各レジスタ表示例

3.1.1 レジスタウインドウのショートカットキー

レジスタウインドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはレジスタウインドウにフォーカスがある場合に適用されます。

キー操作	機能
F3	カーソル位置のレジスタまたはフラグの初期化をします。
F6	レジスタ値のシンボル表示をします。
F7	レジスタ値が指しているメモリの内容を表示します。
F9	レジスタ値を10進数で表示します。
Enter	カーソル位置のレジスタ値またはフラグの値を変更するダイアログボックスを表示します。

3.1.2 レジスタウインドウのローカルメニュー

初期化(Z)	F3
インクリメント(+)	
デクリメント(-)	
値の変更(C)	Enter
CPUレジスタ ▶	
✓ CPUレジスタ表示(Q)	
TLBレジスタ表示(T)	
64ビット表示(O)	

[レジスタ]メニューにはレジスタウインドウの制御に関するコマンドが収められています。

メニュー項目	機能
初期化(Z)	カーソル位置のレジスタ値を0にします。
インクリメント(+)	カーソル位置のレジスタ値をインクリメントします。
デクリメント(-)	カーソル位置のレジスタ値をデクリメントします。
値の変更(C)	カーソル位置のレジスタ値を変更するダイアログボックスを表示します。
CPUレジスタ	CPUレジスタ値の表示方法を設定します。
CP0レジスタ	CP0レジスタ値の表示 / 非表示の切り替えをします。
TLBレジスタ	TLBレジスタ値の表示 / 非表示の切り替えをします。
64ビット表示	レジスタの値の64ビット表示を設定します。

CPUレジスタ

CPUレジスタ	10進数表示(D) F9
	シンボル表示(S) F6
	メモリ表示(M) F7

メニュー項目	機能
10進数表示(D)	レジスタ値を10進数で表示 / 非表示の切り替えをします。
シンボル表示(S)	レジスタ値のシンボルを表示 / 非表示の切り替えをします。
メモリ表示(M)	レジスタ値が指しているメモリ内容の表示 / 非表示の切り替えをします。

3.1.3 レジスタウインドウでのマウス操作

レジスタまたはフラグ部分を左ダブルクリックすると、値変更を行うダイアログボックスが表示されます。

3.2 ブレークウインドウ

ブレークウインドウは、現在設定されているブレークポイントを表示します(図 3)。

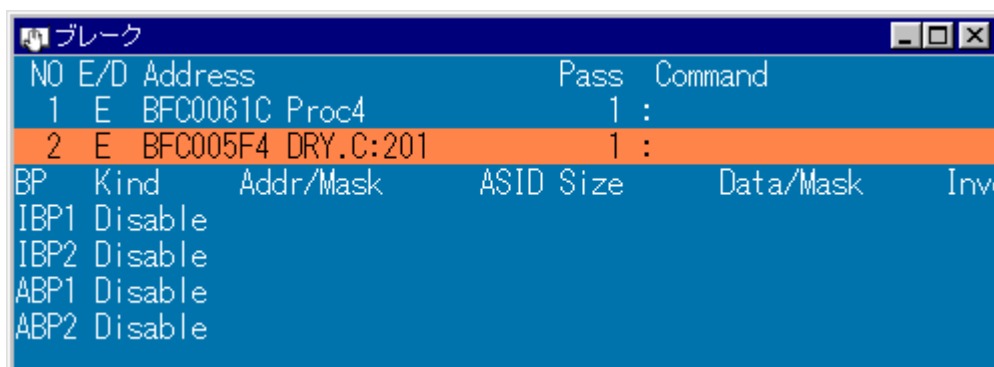


図 3 ブレークウインドウ

3.2.1 ブレークウインドウのショートカットキー

ブレークウインドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはブレークウインドウにフォーカスがある場合に適用されます。

キー操作	機能
F3	全てのブレークポイントを削除します。
F4	現在のブレークポイントの設定を保存します。
F6	F4で保存したブレークポイントの設定を呼び出します。
F7	ハードウェアブレークポイントの設定をします。 (『4.1.1 ハードウェアブレークポイントの設定ダイアログボックス』24 頁参照)
F9 Enter	現在選択されているブレークポイントの許可/禁止を切り替えます。
INS	ブレークポイントの設定ダイアログボックスを表示します。
DEL	現在選択されているブレークポイントを削除します。

3.2.2 ブレークウインドウのローカルメニュー

ブレーク(B)

ブレーク設定(A)..	Ins
禁止(D)	F9,Enter
削除(C)	Del
全て削除(L)	F3
設定のセーブ(S)	F4
設定のロード(L)	F6
ハードウェアブレーク(B)..	F7

[ブレーク]メニューにはブレークウインドウの制御に関するコマンドが収められています。

メニュー項目	機能
ブレーク設定(A)	新規にブレークポイントを設定するダイアログボックスを表示します。
禁止(D)/許可(E)	カーソルのあるブレークポイントの許可 / 禁止の切り替えをします。
削除(C)	カーソルのあるブレークポイントを削除します。
全て削除(L)	全てのブレークポイントを削除します。
設定のセーブ(S)	現在のブレークポイントの設定を保管します。
設定のロード(L)	以前保管されていたブレークポイントの設定をロードします。
ハードウェアブレーク(E)	ハードウェアブレークポイント設定ダイアログが設定されます。(『4.1.1 ハードウェアブレークポイントの設定ダイアログボックス』24頁参照)

3.2.3 ブレークウインドウでのマウス操作

ブレークポイント部分を左ダブルクリックすると、ブレークポイントの有効/無効を切り替えます。

3.3 I/Oウィンドウ

I/Oウィンドウは、指定された内部周辺I/Oレジスタの表示と設定を行います(図 4)。

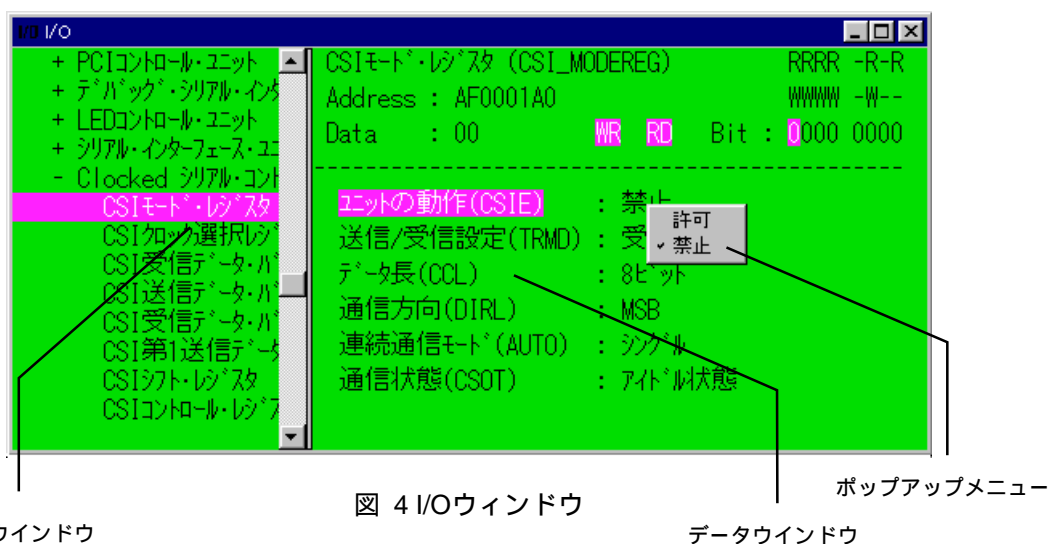


図 4 I/Oウィンドウ

3.3.1 I/Oウィンドウのショートカットキー

I/Oウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはI/Oウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
Tab	アイテム/データウィンドウの切り替えをします。
Enter	アイテムウィンドウの場合、階層構造の表示/非表示をします。 データウィンドウの場合、一つ下階層に移動します。
ESC・BS	データウィンドウの場合、一つ上階層に移動します。
	データウィンドウの場合、設定値を変更します。

3.3.2 I/Oウィンドウのローカルメニュー

I/O

▼ 略称(N) 名称(I)
説明文の表示(H)
定義ファイルの読み込み(L)

[I/O]メニューにはI/Oウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
略称(N)	レジスタ名を表示します。
名称(I)	レジスタの機能名称を表示します。
説明文の表示(H)	レジスタの説明を表示します。
定義ファイルの読み込み(L)	定義ファイル(RPTVR412x.IO)からレジスタの情報を読み込みます。

3.3.3 I/Oウィンドウでのマウス操作

設定値の変更：

変更したいデータの内容をクリックするとポップアップメニューが表示されます。メニュー項目を選択して値の変更、参照をします。

ポップアップメニューは、レジスタによって項目が異なります。

ローカルメニューの表示：

I/Oウィンドウのアイテムウィンドウ内で右ボタンをクリックします。

クリップボードへのコピー：

I/Oウィンドウのアイテムウィンドウ内で、任意の位置で左ボタンを押し、ボタンを押したまま選択領域を指定します。ボタンを放すと選択されていた領域がクリップボードにコピーされます。

4 ダイアログコマンド

PARTNER-VR4120-TPには、ダイアログボックスやツールバーによってデバッグを行う方法と、コマンドウインドウにダイアログコマンドを入力してデバッグを行う方法の2通りあります。

4.1 ダイアログボックス

PARTNER-VR4120-TPに依存したダイアログボックスの説明をします。

4.1.1 ハードウェアブレイクポイントの設定ダイアログボックス

[ブレイク]-[ハードウェアブレイク]

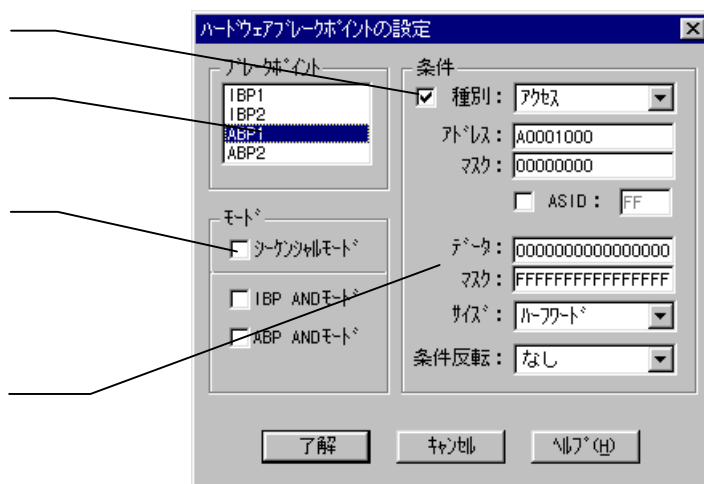
[ハードウェアブレイクポイントの設定]ダイアログボックスは、ハードウェアのブレイクポイントを設定します。

ポイントを有効にするとき
チェックします

設定するポイントを選択し
ます

モードを指定します。シー
ケンシャルモードのときは
ANDモードは無効になりま
す

データとサイズはアクセス
条件の時有効です



ブレイクポイントには、実行条件によるIBP1、IBP2、データアクセス条件によるABP1、DABP2の4ポイントがあります。

モードには次のものがあります。ANDモードのときは、各ポイントが有効である必要があります。

シーケンシャル：IBP1またはABP1の発生後に、IBP2またはABP2が発生したときにブレイクします。

このモードを指定したとき、IBPおよびABPのANDモードは無効になります。

IABP AND：IBP1とIBP2のAND条件でブレイクします。

DABP AND：ABP1とABP2のAND条件でブレイクします。

種別は、データアクセス条件の時の、アクセス種別を指定します。

アドレスは、命令実行アドレス/アクセスアドレス値を指定します。マスクでアドレスのマスク指定をすることができます。

データは、データ値条件を指定します。マスクでデータのマスク指定をすることができます。

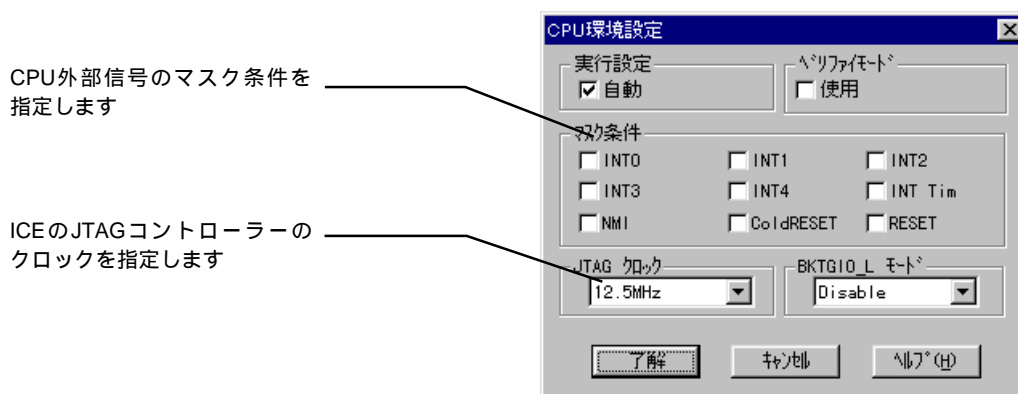
サイズは、データのアクセスサイズ条件を指定します。

条件反転は、アドレス値、データ値の一致条件を反転する場合に指定します。

4.1.2 CPU環境設定ダイアログボックス

[実行]-[CPU環境設定]

[CPU環境設定]ダイアログボックスは、CPUの動作条件を設定します。



CPU外部信号のマスク条件を指定します

ICEのJTAGコントローラーのクロックを指定します

実行設定は、PARTNERおよびICEの内部処理のために、CPUをブレークさせた場合に、再実行を自動で行うか行わないかを選択します。

ベリファイモードは、メモリ書き込みを行う操作やコマンドのときに、書き込み後にメモリベリファイを行うか、行わないかを選択します。ベリファイは、最初のエラーがあったアドレスで終了します。

BKTGIO_Lモードは、BKTGIO_L端子の使用方法を選択します。

4.1.3 TLBの設定ダイアログボックス

[レジスタ]-[値の変更]

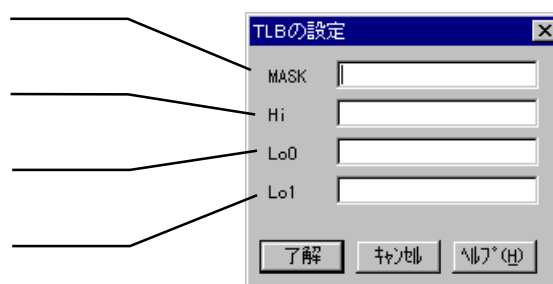
[TLBの設定]ダイアログボックスは、TLBエントリを設定します。

ページマスクデータアドレスを指定します

エントリHiデータを指定します

エントリLo0データを指定します

エントリLo1データを指定します

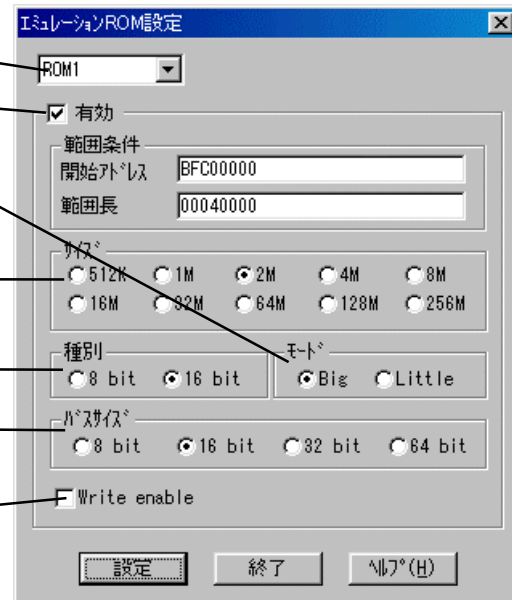


4.1.4 エミュレーションROM設定ダイアログ

[実行]-[エミュレーションROM設定]

[エミュレーションROM設定]ダイアログボックスは、エミュレーションROMの領域を設定します。

- 設定するボードを選択します
- 使用するときにはチェックします
- ROMのエンディアンを指定します
- ROMサイズ(bit単位)を指定します。エミュレーションのハードウェアによって指定できないサイズがあります
- ROM種別を指定します
- バスサイズを指定します
- 書き込み許可のときチェックします



開始アドレスはエミュレートするROMの最下位アドレスを指定し、**範囲長**へは、エミュレートするROMの容量をバイト数(8バイトの境界)で入力します。範囲長に0を指定した場合は、解除(使用しない)と同じになります。16ビットのROMを使用する場合、種別の8bitとサイズの512Kは指定できません。

設定ボタンを押したときにのみ指定した内容が設定されます。**終了ボタン**でダイアログを終了します。

RTE-1000-TPの場合は、**ボードの選択**、Write Enableと**バスサイズ**の64bitは指定できません。

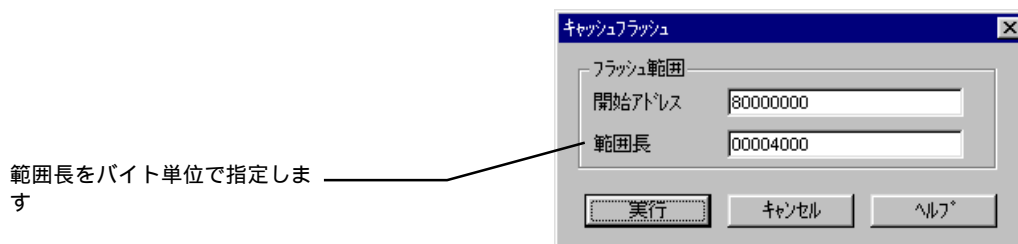
RTE-2000-TPの場合は、バスサイズにより複数枚のボードを使用します。そのため、**ボードの選択**と**バスサイズ**に次の関係があります。記載がないバスサイズは設定できません。

ボード選択	バスサイズ	ボードスロット位置	ボード選択	バスサイズ	ボードスロット位置
ボード1	8 bit	#3	ボード3	8 bit	#5
	16 bit	#3		16 bit	#5
	32 bit	#3+#4		32 bit	#5+#6
ボード2	64 bit	#3+#4+#5+#6	ボード4	8 bit	#6
	16 bit	#4		16 bit	#6

4.1.5 キャッシュフラッシュダイアログボックス

[実行]-[キャッシュフラッシュ]

[キャッシュフラッシュ]ダイアログボックスでは、キャッシュのクリア（フラッシュ）を行います。



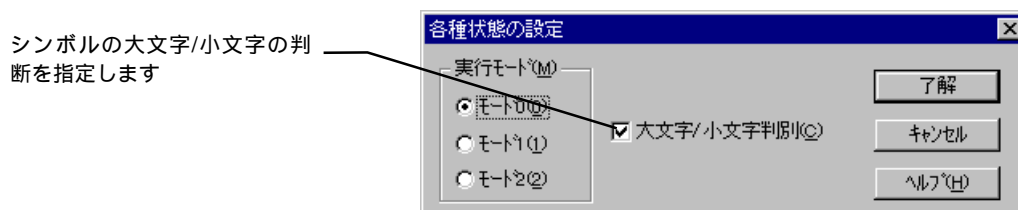
範囲長をバイト単位で指定します

フラッシュは命令、データの両キャッシュラインを無効にします。データキャッシュについてはライトバックをします。

4.1.6 各種状態の設定ダイアログボックス

[設定]-[オプション設定]

[各種状態の設定]ダイアログボックスでは、PARTNER-VR4120-TPの各種状態を設定します。



シンボルの大文字/小文字の判断を指定します

実行モードは現在作用しません。将来の拡張のために用意しています。

実行モードはユーザプログラムの実行時の状態を制御します。PARTNER-VR4120-TPでは、特別な制御を行っていません。デフォルト(モード0)に設定します。

大文字 / 小文字判別は関数名や変数名など登録されたシンボルの大文字 / 小文字の判別をするか否かを指定します。

マーク(チェック)すると大文字 / 小文字の判別をします。マークを解除すると判別を行いません。

マークした場合は、シンボルを使用できる操作に関して、登録されたシンボルと同様に正しく大文字 / 小文字で指定します。正しく指定されていない場合は、

シンボルを参照できません。
マークを解除した場合は、大文字 / 小文字に関係なく指定された文字列に該当するシンボルを参照します。

5 コマンドリファレンス

PARTNER-VR4120-TPのコマンドウインドウに入力可能なコマンドの説明をします。詳しい内容(入力例など)はオンラインヘルプを参照してください。

5.1 コマンド解説の規約

PARTNER-VR4120-TPのコマンドは、コマンド名とパラメータリストによって構成されます。また、パラメータは省略できる場合もあります。省略できるパラメータに関しては鉤括弧([...])で示します。2通り以上の選択要素がある場合には中括弧({...})と|で要素内容を記述します。パラメータが省略された場合は、PARTNER-VR4120-TPの初期値や以前実行したコマンドの続きの値などが用いられます。

5.2 機能別コマンド

5.2.1 ハードウェアの初期化

INIT PARTNER-VR4120-TPと ICE を初期化します。

5.2.2 CPUリセット

RESET ターゲットCPU をリセットします。

5.2.3 環境設定コマンド

```
ENV [[!]]AUTO] [[!]]VERIFY] [[!]]NMI] [[!]]CRESET] [[!]]RESET]
  [[!]]INT0] [[!]]INT1] [[!]]INT2] [[!]]INT3] [[!]]INT4] [[!]]INTT]
  [JTAG{25|12|5M|2M|1M|500K|250K|100K }]
  [{DISABLE|BRKOUT|BRKIN|TRGOUT|TRGIN }]
```

各種入力信号のマスクなどの CPU の各種環境を設定します。

```
ROM [アドレス[範囲]] [ROM8|ROM16][BUS8|BUS816|BUS32]
  [512K|1M|2M|4M|8M|16M|32M|64M|128M|256M][/B|/L]
```

エミュレーション ROM の種類、アドレス範囲を設定します。
RTE-1000-TP 用です。

```
ROM1 ~ ROM4 [アドレス[範囲]] [ROM8|ROM16][BUS8|BUS816|BUS32|BUS64]
  [512K|1M|2M|4M|8M|16M|32M|64M|128M|256M][/B|/L][[!]]WREN]
```

エミュレーション ROM の種類、アドレス範囲を設定します。
RTE-2000-TP 用です。

NC [アドレス 範囲] 非メモリキャッシュの領域を設定します。

NCD リスト 非メモリキャッシュの領域を解除します。

NROM [アドレス 範囲] エミュレーションメモリ割付除外領域の参照と設定します。

NROMD リスト エミュレーションメモリ割付除外領域を解除します。

ACCS [32|64] メリアクセス時のアクセスサイズを設定します。

5.2.4 プログラムロード

L [ファイル名] デバッグプログラムとデバッグ情報を読み込みます。

5.2.5 ファイルリード/ライト

RD ファイル名, 番地 指定ファイルを指定番地に書き込みます。

WR ファイル名, 範囲 範囲指定されたメモリ内容を指定ファイルに書き込みます。

5.2.6 プログラム実行

T [回数]	デバッグプログラムをトース実行します。(F8)
P [回数]	デバッグプログラムをストップ実行します。(F10)
G [=実行番地][,ブレーク番地][,/W]	
G@ [,/W]	デバッグプログラムを実行します。(F5,F7) /W:実行中コマンド禁止
SC	デバッグプログラムを強制ブレークします。(ESC)

5.2.7 ブレークポイント

BP [番地[,回数[,コマンド]]]	ブレークポイントを指定番地に設定します。(F9)
BC リスト	リストで指定されたブレークポイントを解除します。
BD リスト	リストで指定されたブレークポイントを無効にします。
BE リスト	リストで指定されたブレークポイントを有効にします。

5.2.8 ハードウェアブレークポイント

BH	ハードウェアブレークポイントを表示します。
BH {IBP1 IBP2} [アドレス [アドレスマスク]] [反転条件] [[!] AND]	実行条件によるハードウェアブレークポイントを設定します。
BH {ABP1 ABP2} [アドレス [アドレスマスク [データ [データマスク]]]] [種別] [アクセスサイズ] [反転条件] [[!] DAND]	データアクセス条件によるハードウェアブレークポイントを設定します。
BH ![IABP1 IABP2]	指定されたハードウェアブレークポイントを無効にします。
BH ![DABP1 DABP2]	指定されたハードウェアブレークポイントを無効にします。
BH [!] SEQ	シグナルモードの指定をします。

5.2.9 コード表示

V[.][ファイル名:][行]	指定ファイルの指定行をコードウィンドウに表示します。
V 関数名	指定関数のソースファイルをコードウィンドウに表示します。
U [番地]	コードウィンドウに指定番地から逆アセンブル表示を行います。
UPUSH [番地]	現在の表示アドレスをアドレススタック(8段の内部スタック)にPUSHして指定番地から逆アセンブル表示します。
UPOP	最後にUPUSHされたアドレスから逆アセンブル表示してアドレススタックをPOPします。
UEND	最後にUPUSHされたアドレスから逆アセンブル表示します。

5.2.10 レジスタ表示/変更

R	レジスタの値を表示します。
_レジスタ=式	指定レジスタを式の値に変更します。
R レジスタ	レジスタ値の変更をします。

5.2.11 システムレジスタ表示/変更

CP0	CP0 レジスタ表示
CP0 CP0 レジスタ名(番号)=データ	指定 CP0 レジスタを変更します。
TLB [レジスタ番号][,レジスタ番号(E)]	TLB レジスタ表示
TLB レジスタ番号,マスク,エントリHi,エントリLo0,エントリLo1	指定 TLB レジスタを変更します。

5.2.12 メモリ表示/変更

D[型] 範囲[,回数][,基数]	範囲内のメモリ内容を型書式と基数指定にしたがって表示します。
E[型] 番地	指定番地より型指定された形式でメモリ内容を変更します。
F[型] 範囲,リスト	指定範囲を型指定された形式でリストの値でフィル(Fill)します。
S[型] 範囲,リスト	指定範囲を型指定された形式でリストのメモリパターンを検索します。
C 範囲,番地	指定範囲を指定番地と比較(コパア)します。
M 範囲,番地	指定範囲を指定番地にブロック移動します。

5.2.13 I/Oポート入力/出力

PI[型] 番地[,/C]	型指定された形式で指定アドレスの内容を表示します。
PO[型] 番地,データ[,/C]	型指定された形式で指定アドレスにデータを出力します。

5.2.14 シンボル表示/設定

X[シンボル名]	シンボル名(指定がなければすべてのシンボル)を表示します。
[.]名前=アドレス	名前のシンボルを指定アドレスで登録(変更)します。

5.2.15 アセンブル

A 番地	指定番地よりアセンブルしてメモリに直接展開します。
------	---------------------------

5.2.16 バックトレース

K	Cの関数のバックトレース表示を行います。
---	----------------------

5.2.17 キャッシュフラッシュ

CAHE [アドレス[範囲長]]
キャッシュフラッシュを行います。

5.2.18 Cのデータ参照/変更

INS C の式[,関数] C の式を評価してインスペクタウインドウに表示します。(F6,Ctrl+I)
W? C の式 C の式をウォッチウインドウに登録します。(Shift+F7,Ctrl+W)
W[型] 番地[,範囲][,基数]
番地と範囲で指定されたメモリ内容をウォッチウインドウに登録します。
Y リスト リストで指定されたウォッチ行を削除します。
VAL C の式[,関数]
? C の式[,関数] C の式を評価して表示します。

5.2.19 文字列の定義(Cの式評価)

DEF 文字列1 文字列2
#DEFINE 文字列1 文字列2
ミクロプロセッサのための文字列を定義します。
DEF 現在登録されている文字列定義内容の表示をします。
DEF * 現在登録されているすべての文字列定義を無効にします。

5.2.20 システム制御

EXIT PARTNER を終了します。
Q PARTNER を終了します。
HELP ヘルプ表示を行います。
VER PARTNER のバージョン表示をします。
MAP 設定されたメモリマップを表示します。
!! コマンドラインストリの表示をします。
! 文字列 コマンドラインストリの文字列サーチをします。
& スルコマンドへの移行/終了
& スルコマンド スルコマンドの実行

5.2.21 システムコール

SYSC アドレス	アドレスで指定されたアドレスをエンポイントとしてシステムコールを ON します。
SYSC OFF	システムコール機能を OFF します。
SYSC	システムコール機能の状態表示

5.2.22 式の表示

H 式	式の値を 8, 10, 16 進数, ASCII および実数で表示します。
H 式 1, 式 2	式 1 と式 2 の和と差の値を表示します。
PRINTF 書式[, パラメータ]	
PF 書式[, パラメータ]	C の関数の printf() と同様なフォーマット表示をします。

5.2.23 基数変更

N 基数	入力基数を 10 進数または 16 進数に設定します。
------	-----------------------------

5.2.24 ロギング/バッチ

> ファイル名	コマンドウィンドウの表示/入力をファイルに出力します。
>> ファイル名	指定ファイルへのアペンド (APPEND) ロギング出力します。
>	ロギングの中断(ログファイルのクローズ)をします。
< ファイル名	コマンドウィンドウの入力をファイルから行います。ESC キーでバッチ処理の中断が可能です。

5.2.25 オプション設定

OPTION {ON OFF}	大小文字判別をします。(SHIFT+F10) 画面コントロール/その他
CLS	コマンドウィンドウのクリアをします。
HOME	コマンドウィンドウのカーソルをホームポジションへ移動します。
LOCATE X 座標, Y 座標	コマンドウィンドウのカーソルを指定した位置へ移動します。
LALL	マクロでの表示出力指定をします。
SALL	マクロでの表示出力抑制指定をします。
LIST	コマンドウィンドウの表示出力指定をします。
NLIST	コマンドウィンドウの表示出力抑制指定をします。
BEL	ベルを鳴らします。
TIME	現在時刻(時:分:秒)を表示します。
WAIT	一時停止します。
PROMPT 文字	プロンプトを指定された文字に変更します。
*	コメント行指定をします。

5.2.26 マクロコマンド

{ マクロ名	マクロ名でマクロ本体を登録します。
DO{ }WHILE 式	C の do..while 文と同様なマクロコマンド
FOR{ }	C の for 文と同様なマクロコマンド
WHILE{ 式	C の while 文と同様なマクロコマンド
REPEAT{ パラメータ	リピートマクロコマンド
BREAK	マクロから抜け出します。
KILL マクロ名	マクロの削除をします。
MLIST [マクロ名]	マクロの内容を表示します。
MLIST >ファイル名	現在登録されているすべてのマクロを指定ファイルに書き込みます。
< ファイル名	指定されたマクロファイルからマクロ読み込みをします。
IF{ 式	C の if,elseif,else 文と同様な条件制御コマンドです。

5.2.27 フラッシュメモリ管理

ZF Stat[,アドレス]	フラッシュメモリの状態表示
ZF Erase, アドレス	指定のアドレスを含むフラッシュメモリの全領域消去
ZF Flush	未書き込みデータの書き込み
ZF Clear	未書き込みデータの破棄

空白ページ

6 高級言語レベルデバッグ

PARTNER-VR4120-TPでは、Green Hills、GNU(exeGCC)、およびCodeWarriorのC言語をサポートしています。

これらの処理系を使ったROM化についての詳しい内容は各処理系のマニュアルを参考にしてください。ここでは、PARTNER-VR4120-TPが必要とするデバッグ情報を出力する方法を説明します。

PARTNER-VR4120-TPはデバッグ対象となるプログラムを読み込む時、同時にそのプログラムのデバッグ情報を読み込みます。したがって、Cコンパイラあるいはアセンブラからデバッグ情報が正しく生成されていなければ正常なデバッグができません。

6.1 GreenHills C

PARTNER-VR4120-TPはELFファイルフォーマットの実行ファイルを読み込みます。実行ファイルの拡張子は".OUT"です。

6.1.1 コンパイルとリンク方法

C言語で作成したモジュールのコンパイル時には、コマンドラインに-g -XDWARF オプションを付加します。このオプションを付加してコンパイルしたモジュールはソースレベルデバッグが可能です。

さらに、-oオプションで実行ファイルのファイル名を指定する場合は、拡張子を .out にします。

```
> ccwipe -cpu=r4100 -g -XDWARF ソースファイル名 -o実行ファイル名.out
ソースファイル名 ソースファイル名
実行ファイル名 実行ファイル名
```

ビルダーを使用してC言語で作成したモジュールのコンパイル時には、以下に示すビルドオプションを付加してください。

```
:dwarf=true
:debuglevel=plain
:outputname=実行ファイル名.out
```

この結果、ELFファイルフォーマット(.OUT)の実行ファイルが作成されます。

なお、最適化関連のオプションを設定した場合には、正しくデバッグできないことがありますので、最適化なしの設定にしてください。

6.1.2 使用例

1) コンパイル

```
> ccwipe -cpu=r4100 -g -XDWARF -c @cmpfiles
cmpfiles コンパイル対象ソースファイルのリストファイル
(複数のソースファイル名称を指定)
```

この結果、拡張子".o"のオブジェクトファイルが作成されます。

2) リンク

```
>lx -map=chkpro.map -o chkpro.out @chkpro.lx @lnkfiles
```

chkpro.lx リンカに引渡すオプション定義(リンクマップ)ファイル

lnkfiles リンク対象オブジェクトファイルのリストファイル
(複数のオブジェクトファイル名称を指定)

この結果、ELFファイルフォーマット(.OUT)の実行ファイルが作成されます。

3) リンクマップの例

実行プログラムで指定される各セクションをメモリ空間のどの位置に配置するのかを決定します。

```
-m 3 -sec
{
    .text      0x80010000      :
    .rodata   align(8)       :
    .syscall                      :
    .secinfo                      :
    .fixaddr                      :
    .fixtype                      :
    .data     align(8)       :
    .sdabase  align(8)       :
    .sdata                      :
    .rodata                      :
    .sbss     align(8)       :
    .bss      align(8)       :
    .heap     align(8) pad(0x4000) :
    .stack    align(8) pad(0x2000) :
}
}
```

6.1.3 MULTI2000での作成

MULTI2000を使用して作成する場合、次の操作でデバッグ情報を出力してください。

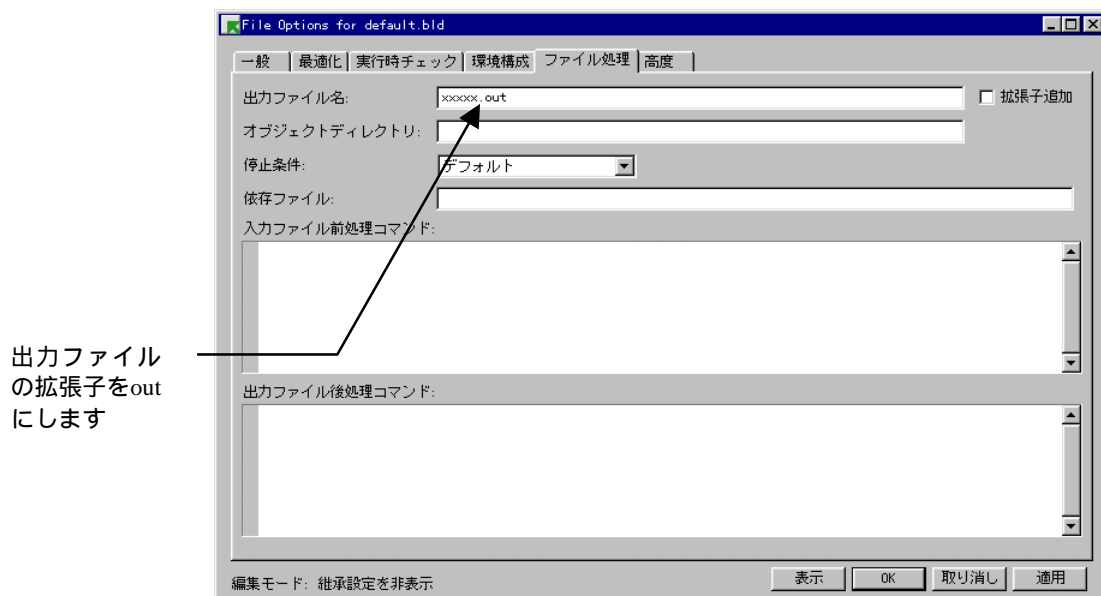
- ProjectメニューからFile Optionダイアログを表示します。
- 高度タブを表示します。
- dualデバッグフォーマット出力を選択します。
- コンパイル対象ドロップリストでアセンブリを選択します。

7 フラッシュメモリのサポート



また、次の操作で出力ファイルの拡張子を".out"にしてください。

- ・ ファイル処理タブを表示します。
- ・ 拡張子".out"のファイル名を指定します。



コマンドラインで使用する場合は、`-G -dual_debug -noobj -o出力ファイル.out`を追加指定してください。

6.2 GNU C(exeGCC)

PARTNER-VR4120-TPはGNU Cの実行ファイルを読み込みます。実行ファイルの拡張子は".OUT"です。

6.2.1 コンパイルとリンク方法

C言語で作成したモジュールのコンパイル時には、-gオプションを付加してください。このオプションを付加してコンパイルしたモジュールはソースレベルデバッグが可能です。さらに、-oオプションで実行ファイルのファイル名を指定する場合は、拡張子を.out にしてください。以下にコンパイル及びリンク時の基本的な指定方法を明記します。

```
>gcc -mips3 -g ソースファイル名 -o実行ファイル名.out -Tコマンドファイル名
ソースファイル名      ソースファイル名
実行ファイル名      実行ファイル名
コマンドファイル名    リンク用コマンドファイル
```

この結果、COFFフォーマット(.OUT)の実行ファイルが作成されます。

なお、最適化関連のオプションを設定した場合には、正しくデバッグできないことがありますので、最適化なしの設定にしてください。

6.2.2 使用例

1) コンパイル

```
>gcc -g -mips3 -c @mpfiles
mpfiles      コンパイル対象ソースファイルのリストファイル
              (複数のソースファイル名称を指定)
```

この結果、拡張子".o"のオブジェクトファイルが作成されます。

2) リンク

```
>ld -Map chkpro.map -Tchkpro.cmd -o chkpro.out @lnkfiles
chkpro.cmd   リンカに引渡すオプション定義(リンクマップ)ファイル
lnkfiles     リンク対象オブジェクトファイルのリストファイル
              (複数のオブジェクトファイル名称を指定)
```

この結果、COFFファイルフォーマット(.OUT)の実行ファイルが作成されます。

3) リンクマップの例

実行プログラムで指定される各セクションをメモリ空間のどの位置に配置するのかを決定します。

```
SECTIONS
{
    _xfer = 0 ;
    _stack_init = 0x8001fff0 ;
    _heap_size = 0x4000 ;
    .text 0x80001000 :
    {
        *(.text)
    }
    .rodata . :
    {
        *(.rodata)
        _erdata = . ;
    }
    .data :
    {
        _data = . ;
        *(.data)
        _edata = . ;
    }
    _gp = . :
    .lit8 . : {
        *(.lit4)
    }
    .sbss . :
    {
        _fdss = . ;
        *(.sbss)
        *(.scommon)
    }
    .bss . :
    {
        *(.bss)
        *(COMMON)
        _end = . ;
        _heap = . ;
    }
}
```

6.3 CodeWarrior C

PARTNER-VR4120-TPはELFフォーマットの実行ファイルを読み込みます。実行ファイルの拡張子は".OUT"です。

【注意】PARTNER-VR4120-TPの起動時に設定するデバッグモード設定オプションには、"-XCW"(Code Warrior Cのデバッグモード)を選択してください。

6.3.1 コンパイルとリンク方法

C言語で作成したモジュールのコンパイルは、ビルド・ツール(CodeWarriorIDE)で行います。

ビルド・ツールでのプロジェクト作成時に、ターゲット設定オプションに関して以下の設定にします。

デバッグフォーマット： DWARF

最適化レベル： 0

さらに、プロジェクトに必要な各種の環境設定(HeapSize, StackSize, Section Mappings...)および必要なモジュールの登録をします。

この結果、ELFフォーマット(.OUT)の実行ファイルが作成されます。

なお、最適化関連のオプションを設定した場合には、正しくデバッグできないことがありますので、最適化なしの設定にしてください。

6.3.2 使用例

1) コンパイルとリンク

ビルド・ツール(CodeWarriorIDE)で行います。

2) リンクマップの例

実行プログラムで指定される各セクションをメモリ空間のどの位置に配置するのかを決定します。

```
MEMORY {
    .text    (RX) :  ORIGIN = 0xa0020000, LENGTH = 0x40000
    .data    (RW) :  ORIGIN = 0xa0060000, LENGTH = 0x20000
    .init    (RW) :  ORIGIN = AFTER(.text, .data), LENGTH = 0x10000
}
SECTIONS {
    ..code :
    {
        _ftext = .;
        *      (.text)
        . = ALIGN(0x8);
        _etext = .;
        *      (.rodata)
        . = ALIGN(0x4);
    } > .text
    ..code_cpp:
    {
        *      (.vtables)          # Only needed for C++
        *      (.exception)       # Only needed for C++
        . = ALIGN(0x4);
        __exception_table_start__ = .;
        EXCEPTION # Only needed in C++
        __exception_table_end__ = .;
    } >> .text
    ..data :
    {
        *      (.data)
        . = ALIGN(0x8);
        _fsdata = .;
        _gp = _fsdata + 0x00008000;
        *      (.sdata)
        . = ALIGN(0x8);
        _fsbss = .;
        *      (.sbss)
        . = ALIGN(0x8);
        _esbss = .;
        _fbss = .;
        *      (.bss)
```

(次頁に続く)


```
        . = ALIGN(0x8);
        _ebss = .;
        *      (COMMON)
} > .data
..tmp :
{
    __static_init = .;
    STATICINIT
    . = ALIGN(8);
} > .init
_stack_addr = . + 0x4000;
_heap_addr = _stack_addr;
_heap_end = _heap_addr + 400*1024;
}
```

空白ページ

7 フラッシュメモリのサポート

PARTNER-VR4120-TPでは、メモリアクセス系のダイアログコマンド（E, F, M, L, RDなどのコマンド）やウィンドウコマンドで、フラッシュメモリを意識することなく書き込みが行えます。

ZFコマンドは、フラッシュメモリの制御や状態表示のためのコマンドです。

実行制御系のダイアログコマンド（G, T, Pなどのコマンド）やウィンドウコマンドでは、フラッシュメモリはROMとして取り扱います。

したがって、ブレークポイントの制御は、ハードウェア・ブレーク制御になります。

7.1 対応デバイス

書き込みに対応するデバイスは、以下の仕様を満たすものに限られます。

- 1 . AMD製またはその互換品で以下の項目に適合すること
 - 1) 単一電源による電氣的消去及び書き込み可能なデバイスであること
 - 2) JEDEC標準コマンドで消去および書き込み制御が可能なこと
 - 3) ブートセクタ以外のセクタサイズが64KBであること
 - 4) ボトムブートタイプのセクタ構成は以下の仕様であること
[16KB x 1セクタ] + [8KB x 2セクタ] + [32KB x 1セクタ] + [64KB x nセクタ]
 - 5) トップブートタイプのセクタ構成は以下の仕様であること
[64KB x nセクタ] + [32KB x 1セクタ] + [8KB x 2セクタ] + [16KB x 1セクタ]
 - 6) セクタ保護(ブロックプロテクト)がなされていないこと
 - 7) CPUに8/16/32ビットのデータバスで接続されていること
 - 8) デバイスの空間がバンク化されていたり一部の空間がマッピングされていないなどの特殊な回路構成でないこと (デバイスの全空間がCPUのメモリ空間にリニアにマッピングされていること)
- 2 . Intel製またはその互換品で以下の項目に適合すること
 - 1) 単一電源による電氣的消去及び書き込み可能なデバイスであること
 - 2) ブートセクタ以外のセクタサイズが64/128/256KBであること
 - 3) ボトムブートタイプのセクタ構成は以下の仕様であること
[8KB x 8セクタ] + [64/128/256KB x nセクタ]
 - 4) トップブートタイプのセクタ構成は以下の仕様であること
[64/128/256KB x nセクタ] + [8KB x 8セクタ]
 - 5) セクタ保護(ブロックプロテクト)がなされていないこと
 - 6) CPUに8/16/32ビットのデータバスで接続されていること
 - 7) デバイスの空間がバンク化されていたり一部の空間がマッピングされていないなどの特殊な回路構成でないこと (デバイスの全空間がCPUのメモリ空間にリニアにマッピングされていること)
- 3 . 上記 1 , 2 仕様とセクタサイズ、セクタ構成のみが異なるデバイス

上記の項目を満たさないデバイスは、ダイアログコマンドやウィンドウコマンドでの書き込みはできませんが、読み込みやユーザプログラムでのアクセスは可能

です。現時点で1と2の仕様を満たすデバイス例を次表に示します。

【対応デバイス】

AMD	Am29F040, Am29F400, Am29F800, Am29LV004, Am29LV400
ST	M29F040, M29F400, M29W008, M29W800
富士通	MBM29F040, MBM29F400, MBM29LV800, MBM29LV008, MBM29LV160
東芝	TC58FVT400, TC58FVB400, TC58FVT800, TC58FVT160
INTEL	28F***C3, 28F***w18, 28F***W30, 28F***K**, 28F***J3*
日立	HMB32S4, HMT32S4
三菱	M***B33****, M***T33****

7.2 環境設定

ダイアログコマンドやウィンドウコマンドで書き込みを行うためには、コンフィギュレーションファイル(RPTVR4120T.CFG)にフラッシュメモリの領域設定が必要になります。設定には、セクタ構成によって2種類 (FLASH_MEM、FLASH_MEM2) があります。

領域設定書式 1 (FLASH_MEM)

『7.1 対応デバイス』の1または2の仕様を満たすデバイスの場合に使用します。ターゲットボードに実装されているデバイスのアドレス、タイプおよびバス幅を設定します。この書式は、FLASH_MEM2の設定と合わせて最大8個まで設定できます。複数定義する場合は昇順に定義してください。

書式： FLASH_MEM *開始アドレス, 終了アドレス, メモリタイプ, CPUバス幅 [, FLASHバス幅]*

*開始アドレスと終了アドレス*には、デバイスがマッピングされているメモリ領域の範囲を指定します。

*メモリタイプ*には、デバイスの種類によって以下の何れかを指定します。

メモリアイプ	デバイスタイプ	セクタ構成
TOP	AMDタイプ	トップブート
BOTTOM	AMDタイプ	ボトムブート
NO	AMDタイプ	ブートブロックなし
I32W_TOP	INTELタイプ、ライトバッファ有	トップブート 32Kword(64KB)/セクタ
I32W_BOTTOM	INTELタイプ、ライトバッファ有	ボトムブート 32Kword(64KB)/セクタ
I32W_NO	INTELタイプ、ライトバッファ有	ブートブロックなし 32Kword(64KB)/セクタ
I32_TOP	INTELタイプ	トップブート 32Kword(64KB)/セクタ
I32_BOTTOM	INTELタイプ	ボトムブート 32Kword(64KB)/セクタ
I32_NO	INTELタイプ	ブートブロックなし 32Kword(64KB)/セクタ
I64_NO	INTELタイプ	ブートブロックなし 64Kword(128KB)/セクタ
I128_NO	INTELタイプ	ブートブロックなし 128Kword(256KB)/セクタ

ライトバッファは、コマンド41Hで制御できるものでなければなりません。

*CPUバス幅*には、CPUとの接続バス幅で8/16/32bitいずれかのバス幅を指定します。たとえば、8bitバスのデバイス2個を組み合わせると16bitバスでCPUに接続した場合は16を指定します。

*FLASHバス幅*には、デバイスのデータバス幅で8/16bitいずれかのバス幅を指定します。省略された場合には、*CPUバス幅*と同じになります。但し、*CPUバス幅*が32の場合には、8または16を必ず指定してください。

領域設定書式 2 (FLASH_MEM2)

『7.1 対応デバイス』の3の仕様を満たすデバイスの場合に使用します。ターゲットボードに実装されているデバイスのアドレス、タイプおよびバス幅を設定します。この書式は、FLASH_MEMの設定と合わせて最大8個まで設定できます。複数定義する場合は昇順に定義してください。

書式： FLASH_MEM2 開始アドレス, メモリタイプ, セクタ情報,
CPUバス幅 [, FLASHバス幅]

開始アドレスは、デバイスがマッピングされているメモリアドレスを指定します。

メモリタイプには、デバイスの種類によって以下の何れかを指定します。

メモリタイプ	デバイスタイプ
0	AMDタイプ
1	INTELタイプ
1001	INTELタイプ、ライトバッファ有 ライトバッファは、コマンド41H で制御できるものでなければなり ません

セクタ情報には、デバイスのセクタ構成を次の書式で指定します。

セクタサイズ:セクタ数,*セクタサイズ*:セクタ数,*セクタサイズ*:セクタ数,...]]

セクタサイズ:セクタ数でデバイスのセクタ仕様を記述します。

セクタサイズ:セクタ数で1セットとして128セットまで指定できます。

セクタサイズ 1セクタのサイズ(16進数)を指定します。

セクタ数 連続するセクタ数(10進数)を指定します。

CPUバス幅には、CPUとの接続バス幅で8/16/32bitいずれかのバス幅を指定します。
たとえば、8bitバスのデバイス2個を組み合わせると16bitバスでCPUに接続した場
合は16を指定します。

FLASHバス幅には、デバイスのデータバス幅で8/16bitいずれかのバス幅を指定し
ます。省略された場合には、CPUバス幅と同じになります。但し、CPUバス幅が
32の場合には、8または16を必ず指定してください。

なお、この書式を定義する場合は、MAP書式(『共通編 3.3章を参照』)と整合性
を保つ必要があり、次の条件を満たさなければなりません。

- 1) MAP書式の後にこの書式を記述すること
- 2) MAP書式で設定した領域内にこの書式で設定する領域が含まれること
- 3) 1つのMAP書式で設定した領域内にこの書式で定義する領域が含まれること

上記の条件を満たさない場合は、PARTNER-VR4120-TPの起動時にエラーメッセ
ージが表示され起動しません。

コンフィギュレーションファイル定義例

正しい設定 1 - 1 : 単一ブロックの設定

```
MAP 00000000, FFFFFFFF          MAP省略時と同じ
FLASH_MEM 03000000,037FFFFFFF, NO, 32, 8
```

正しい設定 1 - 2 : 単一ブロックの設定

```
MAP 00000000, FFFFFFFF
FLASH_MEM2 00000000,0,2000:8,10000:126,2000:8,16,16
```

正しい設定 2 - 1 : 複数ブロックの設定

```
MAP 00000000, 003FFFFFFF
MAP 00400000, 007FFFFFFF
MAP 00800000, 00FFFFFFF
MAP 02000000, 02FFFFFFF
MAP 03000000, 037FFFFFFF
MAP 03C00000, 03FFEFFFF
FLASH_MEM 02000000, 027FFFFFFF, NO, 32, 8
FLASH_MEM 03000000, 037FFFFFFF, NO, 32, 8
```

正しい設定 2 - 2 : 複数ブロックの設定

```
MAP 00000000, 003FFFFFFF
MAP 00400000, 007FFFFFFF
MAP 00800000, 00FFFFFFF
MAP 02000000, 02FFFFFFF
MAP 03000000, 037FFFFFFF
MAP 03C00000, 03FFEFFFF
FLASH_MEM2 00000000,0, 10000:128,32,8
FLASH_MEM2 01000000,0, 10000:128,32,8
```

誤った設定1 : MAP領域外

```
MAP 00000000, 033FFFFFFF
FLASH_MEM 03000000,037FFFFFFF, NO, 32, 8      ←
```

誤った設定 2 : 複数MAPに架かる

```
MAP 00000000, 003FFFFFFF
MAP 00400000, 007FFFFFFF
MAP 00800000, 00FFFFFFF
MAP 02000000, 02FFFFFFF
MAP 03000000, 037FFFFFFF
MAP 03C00000, 03FFEFFFF
FLASH_MEM 02000000, 027FFFFFFF, NO, 32, 8
FLASH_MEM 02800000, 037FFFFFFF, NO, 32, 8      ←
```

7.3 書き込み動作

ダイアログコマンドやウィンドウコマンドで書き込みのための特別な操作は必要としません。通常の操作でフラッシュメモリを意識することなく書き込みできます。

但し、フラッシュメモリはその特性からメモリ内容を書き換えるためにセクタ単位で一度電氣的に消去し、その後書き込みをする必要があります。書き込みには数秒の時間を要します。

したがって、PARTNER-VR4120-TPでは、処理速度の改善や書き込み回数の削減を目的にデバッガ内部に管理バッファを設けて、フラッシュメモリの書き込み動作を以下に示す仕組みで行います。

- 1) ダイアログコマンドやウィンドウコマンドで書き込みが発生
- 2) 該当するアドレスを含むセクタ内容を管理バッファに読み込む
- 3) 書き換えられたアドレスに相当する管理バッファの内容を更新する
- 4) 以下のタイミングで管理バッファをセクタ単位で実際に書き込む

ユーザプログラムの実行開始時

ブロックでの書き込みを伴う処理の終了時

ZFコマンドで明示的に管理バッファをフラッシュした時

その他、内部処理で必要な場合

管理バッファのフラッシュ動作が開始するとコマンドウィンドウやステータスバーに以下のようなメッセージが表示されます。

【メッセージ表示例】

- 1) Flash ROM [42000000-42003FFF] Sect.Erase. /* セクタ消去中 */
表示されているセクタアドレスを消去中です。
- 2) Flash ROM [42000000-42003FFF] Sect.Write. /* セクタ書き込み中 */
表示されているセクタアドレスにデータを書き込み中です。
- 3) Flash ROM [42000000-420FFFFFF] CHIP-Erase. /* チップ全体の消去中 */
表示されているアドレスのチップ全体を消去中です。
- 4) Flash ROM Device Error! /* 消去・書き込み失敗 */
消去または書き込みに失敗しました。環境ファイルの設定間違いや回路不良やデバイス不良が原因です。また、セクタ保護(ライトプロテクト)がなされている場合も書き込みに失敗します。

読み書き動作の技術的補足

フラッシュメモリの読み書き動作を理解するために、デバッガ内部に設けられた管理バッファについて説明します。

管理バッファはフラッシュメモリの容量と同じサイズのメモリが確保され、その状態をセクタ単位で次表のように管理します。

バッファ状態	意味
Invalid	バッファ内のデータは全て無効（バッファ未使用） 初期値
Valid	バッファ内にはフラッシュメモリと同じデータが格納
Dirty	バッファ内のデータが更新されフラッシュメモリには未書き込み

ダイアログコマンドやウィンドウコマンドでフラッシュメモリの領域に対する読み書きが行われると、次のように状態が遷移します。

読み込み要求が発生した場合：

該当セクタの状態がDirtyの場合にのみ、管理バッファからデータを読み込みます。その他の状態では、直接フラッシュメモリから読み込みます。この要求では、状態遷移は発生しません。

書き込み要求が発生した場合：

該当セクタの状態がInvalidの場合にフラッシュメモリのデータを管理バッファに読み込み、状態をValidに変更します。その後、データを管理バッファに書き込み状態をDirtyに変更します。

フラッシュ要求が発生した場合：

Dirtyのセクタを対象にバッファからラッシュメモリへ書き込み、状態をValidに変更します。

クリア要求が発生した場合：

セクタ状態を無条件にInvalidに変更します。この時にDirtyのセクタが存在していてもフラッシュメモリへの書き込みは行いません。

消去要求が発生した場合：

セクタ状態を無条件にInvalidに変更します。この時にDirtyのセクタが存在していてもフラッシュメモリへの書き込みは行いません。

7.4 管理コマンド(ZF)

フラッシュメモリの制御や管理バッファの状態表示に用意されたZFコマンドについて説明します。

ステータス： 書式 ZF Stat[, アドレス]

指定されたアドレスが含まれるデバイスの管理バッファの状態を表示します。アドレスが省略された場合は、全てのデバイスに対する管理バッファの状態を表示します。

イレース： 書式 ZF Erase, アドレス

指定されたアドレスが含まれるデバイス全体の消去をします。アドレスを省略することはできません。

消去を実行すると、管理バッファのセクタ状態を無条件にInvalidにします。Dirtyのセクタが存在した場合、そのデータを破棄します。

フラッシュ： 書式 ZF Flush

Dirtyのセクタを対象に管理バッファからラッシュメモリへ書き込みを行います。書き込みが完了したセクタは、Validにします。

クリア： 書式 ZF Clear

管理バッファのセクタ状態を無条件にInvalidにします。

Dirtyのセクタが存在した場合、そのデータを破棄します。

付録

レジスタ変数

PARTNER-VR4120-TPが使用可能なレジスタ変数は次の通りです。

レジスタ疑似変数	レジスタ
_r0,,,_r31	\$0 レジスタ,,, \$31 レジスタ
_pc	PC レジスタ
_at	\$1 レジスタの別名
_v0,_v1	\$2,\$3 レジスタの別名
_a0,,,_a3	\$4,,, \$7 レジスタの別名
_t0,,,_t7	\$8,,, \$15 レジスタの別名
_s0,,,_s7	\$16,,, \$23 レジスタの別名
_t8,_t9	\$24,\$25 レジスタの別名
_k0,_k1	\$26,\$27 レジスタの別名
_gp	\$28 レジスタの別名
_sp	\$29 レジスタの別名
_s8	\$30 レジスタの別名
_ra	\$31 レジスタの別名

```
>while{ _R0!=_R1      /* R0レジスタとR1レジスタの内容比較 */  
? T                  /* トレース実行コマンド */  
?}                   /* マクロ終わり */  
>
```

上記の例では\$0と\$1のレジスタ値が同じになるまでT(トレース)コマンドを実行します。レジスタと同名のシンボルがある場合にも、レジスタが優先されます。

エラーメッセージ

PARTNER-VR4120-TPのエラーメッセージについて以下に説明します。

コマンドエラー

指定されたコマンドがPARTNERの内部コマンドおよびマクロコマンドとして認識できません。

アドレス指定が誤りです

アドレス入力部に不適当なアドレスが指定された場合や、スタートアドレスとエンドアドレスが逆転している場合に発生します。また、シンボル登録されていないシンボル名が使用された場合にも発生します。

データ指定が誤りです

データ入力部に不適当なデータが指定された場合や、範囲外のデータが指定された場合に発生します。

コマンドのフォーマットが不正です

コマンドの入力書式、パラメータの指定方法や数に間違いがあります。

Verifyエラー

メモリに正しくデータが書き込めませんでした。メモリの実装されていないアドレスやROM領域に書き込み動作をしたときに発生します。

マクロ内でのマクロの定義はできません

マクロの定義(登録)は、PARTNERのコマンドレベルで行ってください。マクロコマンド内での定義はできません。

マクロ内でのマクロ削除はできません

マクロの削除は、PARTNERのコマンドレベルで行ってください。マクロコマンド内での削除はできません。

マクロ名が内部コマンドと重複しています

マクロコマンド定義を行おうとしたマクロコマンド名がPARTNERの内部コマンドと重複しています。違うマクロコマンド名で定義してください。

マクロバッファがいっぱいです

マクロコマンド定義のためのバッファがいっぱい、または定義したマクロコマンドの数が多すぎます。RPTSETUPを使用してマクロバッファサイズを広げてPARTNERを起動してください。

マクロ定義が正しく終了していません{ }

マクロコマンドの定義で中括弧({ })の数が一致していません。

マクロが16レベル以上入れ子になっています

マクロの入れ子(ネスティング)は最大15レベルまで許されます。

指定された設定がありません

ブレークポイント、ウォッチ等の指定番号の設定が存在しません。

ブレークポイントが設定できません(最大15ポイント)

ブレークポイントを15点越えて設定しようとした。ブレークポイントは最大15点まで指定できます。

ウォッチの設定ができません

ウォッチを16点越えて設定しようとした。ウォッチ設定は最大16点まで指定できます。

ウォッチの指定が重複しています

Wコマンドでウォッチ登録しようとした内容はすでにウォッチ登録されています。

ファイルフォーマットが不正です

ロードしようとした実行形式のファイルのフォーマットが異常です。RPTSETUPで指定したデバッグモードとロードしようとするファイルが一致していません。

指定ファイルがありません

コマンド行で指定されたファイルが見つかりません。

指定ファイルがオープンできません

指定されたファイルがオープンできません。そのファイルが存在するか確認してください。また、ファイルをオープンしすぎている場合にも発生します。

ファイルがクリエートできません

ディスクがいっぱいか、ファイルを同時にオープンしすぎています。不要なファイルをクローズするなどしてください。

ディスクがいっぱいです

指定されたディスクの空き容量が足りません。

デバッグ情報がありません

Lコマンドでプログラムをロードするとき、デバッグ情報がプログラムファイル内がありませんでした。

デバッグ情報のフォーマットが不正です

デバッグ情報のフォーマットが不正です。RPTSETUPで指定したデバッグモードとロードしようとするファイルを確認してください。

デバッグ情報領域がいっぱいです(起動時の-Bオプション参照)

デバッグ情報登録用のバッファに空き領域がありません。RPTSETUPで指定したデバッグ情報バッファサイズを拡大してください。

ローカルシンボルの設定はできません

既に登録されているローカルシンボルと同名のグローバルシンボルを登録しようとしています。

Cの変数が見つかりません

?コマンドやVALコマンドで指定したCの変数が見つかりません。

Cの式計算エラー

?コマンドやVALコマンドでのCの式でエラーがあります。

副作用のある演算子は使用できません

?コマンドなどで副作用のある演算子(=, +=, -=等)は使用できません。副作用のある演算子はVALコマンドで使用してください。

PARTNERの内部エラー

PARTNERの内部処理で不都合が生じた場合に表示されます。このエラーは通常起こらないように設計されています。このエラーが発生し、かつ再現性がある場合には、販売会社にご連絡ください。

ターゲット実行中は使用できません

指定したコマンドは、ユーザプログラム実行中には使用できません。ブレイクしてからコマンドを実行してください。

モニタプログラムが正しく実行できません

CPUを制御するためのモニタプログラムが実行できないか、正常に終了しない状態です。初期化コマンドを実行してください。初期化コマンドの実行でこのエラーが表示される場合はターゲットハードウェアのCPUリセットした後、再度初期化コマンドを実行してください。

空白ページ

索引

.

.CFG..... 54, 55

C

Code Warrior C..... 47

コンパイルとリンク方法..... 47

使用例..... 48

F

FLASH_MEM..... 54

環境設定..... 54

FLASH_MEM2..... 55

環境設定..... 55

G

GNU C..... 45

コンパイルとリンク方法..... 45

使用例..... 45

Green Hills C..... 41, 42

コンパイルとリンク方法..... 42

使用例..... 42

I

I/Oウインドウ..... 21

ショートカットキー..... 21

マウス操作..... 22

ローカルメニュー..... 22

INIT.MCR.....8

M

MULTI2000での作成.....43

P

PARTNER-VR4120-TPの環境設定.....4

あ

エラーメッセージ.....62

か

起動.....3

PARTNER-VR4120-TPの環境設定.....4

起動時のエラーメッセージ.....9

初期設定コマンド.....7

起動オプションの設定.....4

@オプション.....6

-Bオプション.....4

-Dオプション.....5

-Eオプション.....5, 6

-SDオプション.....5

-TABオプション.....5

-Xオプション.....5

高級言語レベルデバッグ.....41

Code Warrior C.....47

GNU C.....45

Green Hills C.....42

コマンドリファレンス	29
機能別コマンド	31
CPUリセット	31
Cのデータ参照/変更	36
I/Oポート入力/出力	35
アセンブル	35
オプション設定	38
環境設定コマンド	31
基数変更	37
キャッシュフラッシュ	36
コード表示	34
式の表示	37
システムコール	37
システム制御	36
システムレジスタ表示/変更	34
シンボル表示/設定	35
ハードウェアの初期化	31
ハードウェアブレイクポイント	33
バクトレース	35
ファイルリード/ライト	32
フラッシュメモリ管理	39
ブレイクポイント	33
プログラム実行	33
プログラムロード	32
マクロコマンド	38
メモリ表示/変更	35
文字列の定義(Cの式評価)	36

レジスタ表示/変更	34
ロギング/バッチ	37
コマンド解説の規約	30
コンフィギュレーションファイル	54, 55

さ

ショートカットキー	
I/Oウインドウ	21
ブレイクウインドウ	19
レジスタウインドウ	17
初期設定コマンド	
INIT.MCRでの初期化例	8
初期化で使用するコマンド	7

た

ダイアログコマンド	23
ダイアログボックス	24
CPU環境設定	25
TLBの設定	25
エミュレーションROM設定	26
キャッシュフラッシュ	27
ハードウェアブレイクポイントの設定	24
各種状態の設定	27
チャイルドウインドウ	15
I/Oウインドウ	21
ブレイクウインドウ	19
レジスタウインドウ	16

は

はじめに.....	1
必要なセットアップ.....	1
必要なハードウェア.....	1
フラッシュメモリのサポート	
環境設定.....	54
管理コマンド.....	60
イレーズ.....	60
クリア.....	60
ステータス.....	60
フラッシュ.....	60
書き込み動作.....	58
対応デバイス.....	52
フラッシュメモリのサポート.....	51
ブレイクウインドウ.....	19
ショートカットキー.....	19
マウス操作.....	20
ローカルメニュー.....	20
付録.....	61
エラーメッセージ.....	62

レジスタ変数.....	61
-------------	----

ま

マウス操作	
I/Oウインドウ.....	22
ブレイクウインドウ.....	20
レジスタウインドウ.....	18
メインウインドウ補足.....	12
メニュー.....	13
実行メニュー.....	13

ら

レジスタウインドウ.....	16
ショートカットキー.....	17
マウス操作.....	18
ローカルメニュー.....	17
レジスタ変数.....	61
ローカルメニュー	
I/Oウインドウ.....	22
ブレイクウインドウ.....	20
レジスタウインドウ.....	17

空白ページ

PARTNER ユーザーズ・マニュアル
VRシリーズ

『VR4120-TP個別編』

第一版 発行日 2002年1月

株式会社 マイダス・ラボ

Copyright 2002 Midas lab Inc./Kyoto Micro Computer Co.,LTD.