

KIT-V850E2/MN4-TP-H

ユーザーズ・マニュアル (Rev. 1.02)

RealTimeEvaluator

■ ソフトウェアのバージョンアップ

- 最新のRTE for Win32 (Rte4win32)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/rte4win_32.htm

■ ご注意

- KIT-V850E2/MN4-TP-H(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。
- 本製品は万全の注意を持って作製されていますが、株式会社マイダス・ラボは当該製品について、不具合が内在していないことを保証しません。
- 本製品をご利用になった結果については、販売会社、及び、株式会社マイダス・ラボは一切の責任を負いません。
- 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。

■ 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

Rev. 1.00	2010.03.20	初版
Rev. 1.01	2010.06.24	仕様変更に伴う修正 <ul style="list-style-type: none">・2章 ハードウェア仕様のブレーク機能・7章 ブレークポイントの説明変更デュアルコアでの注意事項を追加
Rev. 1.02	2010.11.08	<ul style="list-style-type: none">・付録.B abpコマンドのseq12, seq34, exec^hラメータを削除 付録.B optbyteコマンドを追加

目次

1. はじめに.....	4
2. ハードウェア仕様.....	5
3. RTE FOR WIN32の設定.....	7
CHKRTE2. EXEの起動.....	7
4. 初期設定コマンド.....	10
MULTIデバッガを使用する場合.....	10
5. インターフェース仕様 (PB-NEXUS-N38).....	11
ピン配置.....	11
6. 内蔵フラッシュROM・サポート機能.....	12
ブロック管理.....	12
キャッシュの仕組み.....	12
フラッシュROMへの書き込み.....	12
INITコマンド.....	12
自動書き込みモードの変更.....	12
キャッシュ制御コマンド.....	13
注意事項.....	13
7. 注意事項.....	14
操作上の注意.....	14
ブレークポイント.....	14
トレース.....	15
実行中のトレース表示.....	15
トレースの中止.....	15
内蔵フラッシュROMへの書き込み.....	15
同時ブレーク／実行、同期ブレーク.....	15
特定レジスタのシーケンスアクセス.....	15
ブレーク中のタイマ動作.....	15
レジスタ.....	16
INITコマンド.....	16
HALT命令.....	16
その他.....	16
付録. A トレース機能の詳細.....	17
トレースの概要.....	17
ディレイカウンタ.....	18
トレースの実行モード.....	18
サブスイッチ、セクション、クォリファイ.....	19
トレースの開始.....	19
トリガ条件.....	20
トレースの停止.....	21
トレースの終了.....	21
強制ディレイモード.....	22

1. はじめに

KIT-V850E2/MN4-TP-HはNEC製RISCマイコン:V850E2/MN4を搭載したシステムをインサーキットエミュレーションするためのキットです。

本製品には下記のものが付属します。

1. RTE for Win32 Set Up Disk	1枚
2. ライセンスシート	1枚
2. ユーザーズマニュアル	1冊

本KITをご使用になるためには、以下のハードウェアが必要です。

- ・ ICE本体 : RTE-2000H-TP-x-x
- ・ ポッド : PB-NEXUS-N38

2. ハードウェア仕様

エミュレーション部

対象デバイス	V850E2/MN4	
使用するRTE-TPの形式	RTE-2000H-TP + PB-NEXUS-N38	
エミュレーション機能		
動作周波数(*8)	100KHz~	
インターフェース	Nexus仕様	
動作電圧	3.3V	
JTAG-CLK	10KHz - 25MHz	
イベント機能		
イベント数		
実行アドレスの設定	8	
データアクセスの設定	6	
DMAサイクルの設定	6	
条件	アドレス指定	範囲指定可
	データ指定	範囲指定可
	ステータス指定	R, W, R/W
他	シーケンシャル器段数	4
	パスカウンタ	12ビット
ブレーク機能		
H/Wブレークポイント		
アクセス系ブレークポイント*6	4	
	アドレス指定	Mask指定可
	データ指定	Mask指定可
	ステータス指定	R, W, R/W
S/Wブレークポイント	100 *5	
イベントによるブレーク設定	可	
ステップブレーク	可	
マニュアルブレーク	可	
外部信号によるブレーク (High/Low edge)*7	可	
トレース機能		
トレースデータバス	8bit	
トレースデータ有効メモリ	1M-word	
トリガ設定		
実行アドレスによるトリガ設定*6	可	
データアクセスによるトリガ設定*6	可	
イベントによるトリガ設定	可	
外部入力によるトリガ設定	可	
開始、停止指定(サブスイッチ)*6	可	
トレースディレイ	0 - FFFFF	
トレースクロック	DDR-133MHz (max)	
タイムタグ	100nS - 30h	
逆アセンブルトレース表示機能	有	
完全トレースモード指定機能 (no real time)	有	
外部ROMエミュレーション機能(*4)		
ブロック内マップ機能 (USER/EMEM)	64K-Word	
RAMとして使用	可	
メモリ容量	8M - 128Mバイト	
アクセスタイム ()内はバーストサイクル時	35nS(30nS) (*1)	
動作電圧	1.8V - 5V (*2)	
電氣的条件	LV-TTL, 5Vトリアント (*3)	

エミュレーション可能なROM数	
DIP-32pin-ROM (8bit-ROM)	4 (max)
DIP-40/42pin-ROM (16bit-ROM)	4 (max)
拡張16BIT-標準ROMコネクタ	4 (max)
エミュレーション可能なROMの容量 (bit)	
DIP-32-ROM (8-bit bus)	1M, 2M, 4M, 8M (27C010/020/040/080)
DIP-40-ROM (16bit-bus)	1M, 2M, 4M (27C1024/2048/4096)
DIP-42-ROM (16bit-bus)	8M, 16M (27C8000/16000)
拡張16bit-標準ROM (16bit-bus)	1M, 2M, 4M, 8M, 16M, 32M, 64M, 128M, 256M (32M ^{ハイト})
バス幅指定 (bit)	8/16/32
端子マスク機能	STOP, WAIT-, HLDRQ, RESET-
実行時間計測機能 (内はJTAGCLK=25MHz時の値)	
分解能 (nS)	t = JTAGCLK周期の2倍 (80nSec)
最大計測時間 (nS)	t * 2 ³¹ (約171Sec)

- *1, 2, 3. RTE-2000H-TP+CBL-STD16-2Kを使用した場合の値です。
- *2. 2.3V以下で使用する場合は各ケーブルのDC特性に注意ください。電氣的に整合しない場合があります。
- *4. ユーザシステム上にROMエミュレーションケーブルを接続するためのROMソケット、または専用のコネクタが必要です。詳しくはRTE-2000H-TP本体のマニュアルを参照ください。
尚、RTE-2000H-TPでは、E.MEM基板を最大4枚まで実装でき、その時の最大容量は128M^{ハイト}です。
- *5. ROM領域内へのソフトウェア・ブレークポイントの最大設定数は8点までです。
- *6. 以下は同一の資源をシェアしています。
・アクセス系ブレークポイント
・実行アドレス、及びデータアクセスによるトレーストリガ
・サブスイッチの通過条件
- *7. 外部信号のブレークを有効にするためには、EVTI-信号を占有します。
- *8. 100KHz以下での動作をご希望の場合は、別途ご相談ください。

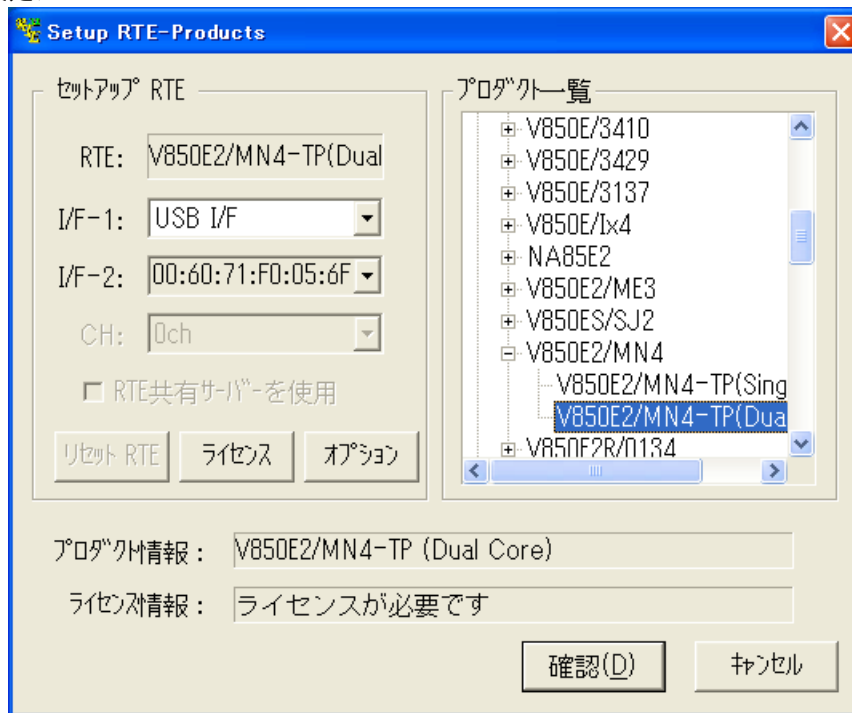
3. RTE for WIN32の設定

『RTE for WIN32』の設定について説明します。

ChkRTE2.exeの起動

ユーザシステムとの接続を完了し、全ての機器の電源が投入された状態で ChkRTE2.exe を起動し、『RTE for WIN32』の環境設定を実施してください。『RTE for WIN32』の環境設定は、新規にハードウェアを設置した時に必ず1回は実施してください。

<RTEの設定>



<RTEの選択>

プログラク一覧より、TPの下層にある **V850E2/MN4-TP(SingleCore)**、**V850E2/MN4-TP(DualCore)** のどちらかを指定してください。

V850E2/MN4-TP(SingleCore) : シングルコアのマイコンの場合に指定

V850E2/MN4-TP(DualCore) : デュアルコアのマイコンの場合に指定

<I/F-1, I/F-2の選択>

使用するホストインターフェースに合ったものをプルダウンメニューから選択して指定してください。(画面は、USB-I/Fを割り付けた場合です)

<ライセンス>

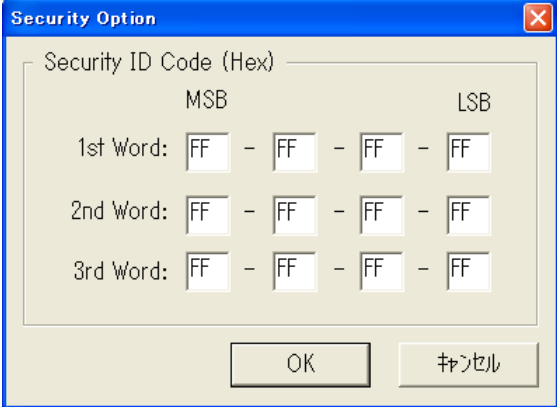
ボタンをクリックして、KITに添付のライセンス設定シートを見て、ライセンスの設定を行ってください。詳細は、『RTE for WIN32』のマニュアルを参照してください。



rte4win32は、Ver. 7.04.06 ~をご使用ください。

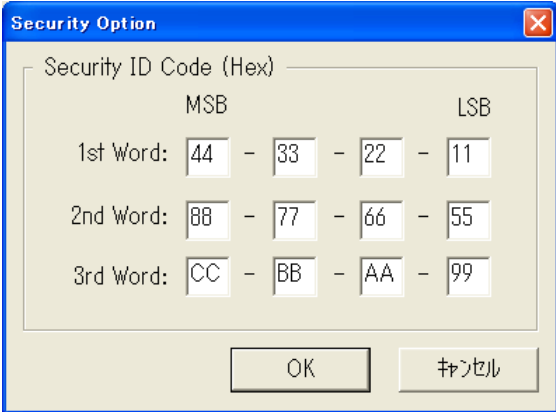
<オプション>

このボタンをクリックして、セキュリティIDコードを入力してください。



<初期状態>

以下は、IDコードの入力例です。IDコードを変更した場合、このオプション画面で必ず同じ値に変更してください。



<入力例>



内蔵ROM/フラッシュ・メモリのセキュリティ機能について

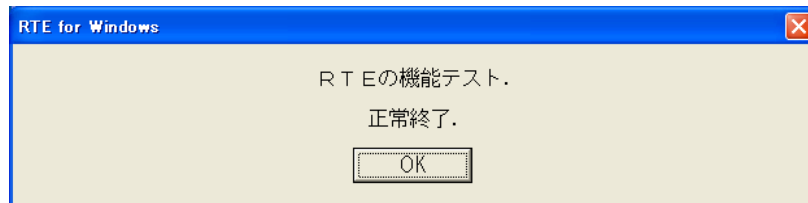
デバッグを起動するためには、12バイトのIDコード認証が必要です。

IDコードについては、GPUのマニュアルをご参照ください。

- ・当該オプションで入力したIDコードとGPUのIDコードが一致しなければデバッグは起動しません。

<機能テスト>

機能テストは、ユーザシステムとの接続が正しく行われ、IDコードの認証がパスできる状態になっていることが必要です。RTEの設定後、画面の指示に従い機能テストを実施すると、正常終了時に下記のダイアログが表示されます。この状態になれば、デバッガからの制御が可能です。



途中でエラーになる場合は、ユーザシステムに障害があるか、JTAG-IFケーブルが正しく接続できていない可能性がありますので、それらの確認を行ってください。



CHKRTE2.EXEの機能テストは、RTE-2000H-TPとユーザシステムが接続され、両方に電源が入っている状態で行ってください。

4. 初期設定コマンド

デバッグを開始する前に、ユーザシステムのハードウェアに依存した初期設定が必要です。初期設定のためのコマンドとして以下が用意されていますので、必要に応じて、正しく設定してからご使用ください。

MULTIデバッガを使用する場合

ターゲットウインドウ内で以下の内部コマンドを使用します。

ENVコマンド

- ・ 端子マスクの指定
- ・ JTAGクロックの指定
- ・ その他

IFROMENV

- ・ 内蔵フラッシュROMの書込みに関する設定

ROMコマンド

- ・ ROMのエミュレーション条件の指定

NC/NCDコマンド

- ・ デバッガ内のキャッシュ機能領域の指定

NSPB/NSPBDコマンド

- ・ ソフトブレーク禁止領域の指定

NROM/NROMDコマンド

- ・ 強制ユーザ領域の指定

5. インターフェース仕様 (PB-NEXUS-N38)

ピン配置

Pin番号	接続信号名	入出力 (User Side)	処理 (User Side)
1	GND	---	GNDに接続
3	TCK	Input	4.7K - 10K Ω プルアップまたはプルダウン
5	TMS	Input	4.7K - 10K Ω プルアップまたはプルダウン
7	TDI	Input	4.7K - 10K Ω プルアップまたはプルダウン
9	TDO	Output	22 - 33 Ω シリーズ抵抗 (推奨)
11	MSE00	Output	10 - 22 Ω シリーズ抵抗 (推奨)
13	MSE01	Output	10 - 22 Ω シリーズ抵抗 (推奨)
15	OPEN	---	
17	MCKO	Output	10 - 22 Ω シリーズ抵抗 (推奨)
19	OPEN	---	
21	MDO[0]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
23	MDO[1]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
25	MDO[2]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
27	MDO[3]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
29	MDO[4]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
31	MDO[5]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
33	MDO[6]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
35	MDO[7]	Output	10 - 22 Ω シリーズ抵抗 (推奨)
37	GND	---	GNDに接続

Pin番号	接続信号名	入出力 (User Side)	処理 (User Side)
2	GND	---	GNDに接続
4	VCC10	Output	+3.3Vの電源に直結 (電圧センス用)
6	TRST-	Input	4.7K - 50K Ω プルダウン
8	RESET-	BIDIR	CPUのRESETへ接続 (OC. 出力)
10	FLMDO	Input	4.7K - 50K Ω プルダウン
12	RDYZ (RDY-)	Output	22 - 33 Ω シリーズ抵抗 (推奨)
14	EVTO	Output	22 - 33 Ω シリーズ抵抗 (推奨)
16	EVTI	Input	4.7K - 50K Ω プルアップ
18	OPEN	---	
20	OPEN	---	
22	---	Output	GNDへの処理を推奨します。
24	---	Output	GNDへの処理を推奨します。
26	---	Output	GNDへの処理を推奨します。
28	---	Output	GNDへの処理を推奨します。
30	---	Output	GNDへの処理を推奨します。
32	---	Output	GNDへの処理を推奨します。
34	---	Output	GNDへの処理を推奨します。
36	---	Output	GNDへの処理を推奨します。
38	GND	---	GNDに接続

備考： 入出力 (User Side) はユーザ基板側での方向です。

6. 内蔵フラッシュROM・サポート機能

本キットでは、内蔵フラッシュROMに対し自動書き込みをサポートしています。これはフラッシュROM領域に対するデバッガからの書き込み（ダウンロードやその他のメモリライト操作）を一旦ホスト上のメモリに保存（キャッシュ）しておき、実行コマンドが発行された時点で、フラッシュROMの内容と異なる部分に対し、自動的に書き込みを行い、実行を開始する機能です。

この機能を禁止する為には、“ifromenvコマンド”で禁止の設定をしてください。

本機能の詳細を以下に説明します。

ブロック管理

フラッシュROMは複数のブロック (4K) で構成されているため、このブロックを1単位として管理し、各ブロックの状態を次の3つに分類します。

- ・ INVALID : キャッシュ（保存）していない状態 (=初期状態)
- ・ VALID : キャッシュしている内容とフラッシュROMの内容が同じ状態
- ・ DIRTY : キャッシュしているが、実際のフラッシュROMの内容とは異なる状態

キャッシュの仕組み

書き込みデータ、及びフラッシュROMの内容は次の条件でキャッシュされます。

- ・ メモリリード時
フラッシュROM領域をリードした時、該当ブロックがINVALIDの場合にフラッシュROMからリードして保存します。状態はVALIDになります。該当ブロックがINVALID以外のときは、キャッシュしたデータを表示します。
- ・ メモリライト（ダウンロードやその他のライト操作）時
フラッシュROM領域へのライトは、該当ブロックがINVALIDの場合一旦フラッシュROMから読み出した値をキャッシュに入れ、その後、キャッシュに対してライトデータを変更します。このとき、ライトデータとキャッシュデータが一致した場合はVALIDに、一致しない場合はDIRTYになります。

フラッシュROMへの書き込み

実際にフラッシュROMへ書き込みを行うのは、プログラムを実行 (RUN, STEP) するときです。

- ・ 実行に先立ち、DIRTYのブロックを書き込みます。
- ・ 書き込みは、ERASE → WRITE → 内部VERIFY (→ READ VERIFY) で行います。
- ・ 書き込み後、状態はVALIDになります。

INITコマンド

INITコマンドを行うとキャッシュ状態はすべてINVALIDになります。この時、書き込みは行いません。

自動書き込みモードの変更

ifromenvコマンドで設定を変更した場合は、キャッシュの状態はすべて一旦INVALIDになります。この時、書き込みは行いません。

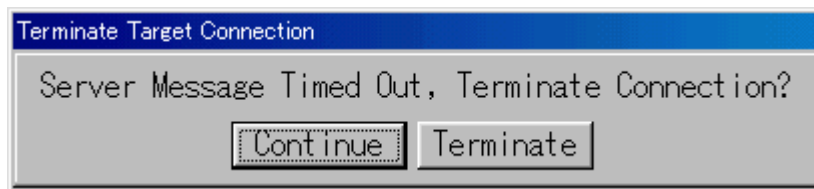
キャッシュ制御コマンド

キャッシュの状態を強制的に変更する以下の2つのコマンドが用意されています。

- ・ ifromclear : キャッシュの状態を全てINVALIDにします。
この時、書き込みは行いません。
- ・ ifromflush : DIRTYのブロックをフラッシュROMに強制的に書き込みます。
書きこみ後の状態は、VALIDです。
- ・ ifromrefill : フラッシュROM領域を全領域読み出し、キャッシュし直します。
全てVALIDになります。

注意事項

- ・ キャッシュがDIRTY状態でプログラムを実行した場合、実際に実行が始まるまでに20秒程度待たされる場合があります。これがフラッシュROMへの書き込み時間です。
- ・ この時、MULTI (デバッガ) が以下の確認メッセージを表示する場合があります。



表示された場合、**Continue**をクリックして続行させてください。



MULTIで以下を設定することで上記タイムアウトのメッセージを表示するまでの時間を延ばすことができます。
SERVERTIMEOUT=30

7. 注意事項

KIT-V850E2/MN4-TPを使用するにあたり、注意して頂く事項を以下にまとめます。

操作上の注意

- 1) ICE本体の電源が切れている状態で、ユーザシステムの電源を入れないでください。双方の故障の原因となります。
- 2) 当該ICEは、CPU内部のデバッグ制御回路(DCU)を外部から制御するものです。そのため以下の条件が満たされていない場合、正しく動作しません。
 - * ユーザシステムとJTAG-IFケーブルが接続されていること。
 - * ユーザシステムの電源が投入され、CPUが正しく動作できる状態にあること。
 - * IDコードの設定が正しく行われていること。(3. RTE for WIN32の設定参照)
- 3) MULTIデバッガを使用してデュアルコアのCPUをICEする場合のサーバ起動方法は以下です。


```
connect rteserv2 -cpu PE1 -cpu PE2 -multicore
```

 なお、次のようにデュアルコアのCPUを、PE1のみで起動しないでください。


```
connect rteserv2 -cpu PE1 -multicore
```

 特定レジスタのシーケンスアクセスを妨害する可能性があります。
 ※シングルコアのマイコンを起動する場合は、rteserv2をパラメータなしで起動してください。



MULTIデバッガは、V5.1.6c ~ をご使用ください。

ブレークポイント

- ・ ROM領域へ設定できるブレークポイントの最大数はEXTBRK機能による8ポイントです。この8点はソフトウェア・ブレークポイントとして扱われます。
- ・ MULTIデバッガのソースウインドウ上からROM領域へ設定するブレークポイントは、EXTBRK機能を使用したブレークポイント(8点)が使用されます。ROM領域には、これ以上の数のブレークポイントは設定できません。



MULTIデバッガ使用時の注意事項

MULTIデバッガでは暗黙的に以下でブレークポイントが使用されます。

1. ソースレベルでのStep, Next, Return, Come等の実行時：2点
2. ソースレベルでダウンロード直後の実行開始時：1点
3. syscall機能使用時：1点(常時)

ROMに配置したプログラムをデバッグする場合は、これらを含めてブレークポイントの設定数の上限を越えないように配慮する必要があります。設定数が越えた場合、以下のエラーが表示されます。

(0x89) Failure to write software breakpoint

- ・ RAM領域へのブレークポイントの設定はソフトウェア・ブレークポイント(ブレーク命令への書き換え)が使用されます。デュアルコア・マイコンでは、このソフトウェア・ブレークポイントは、どちらのコアから設定されたブレークポイントであっても、そのアドレスを実行すればブレークします。しかし、設定したコア以外のコアでブレークした場合、ICEは自動的に継続実行を行ないますので、デバッガからは見かけ上、ブレークしたことはわかりません。
- ・ デュアルコアでソフトウェア・ブレークポイントを設定する時に他方のコアが実行している場合、ICEはそのコアを一時的にブレークし、ソフトウェア・ブレークポイントを設定した後、すぐに再開する処理を行います。(ICE内部で自動的に行われる処理です)。

トレース

MULTIデバッガでトレースを行なう方法として、Multiのトレース機能(TimeMachine)を使う方法とターゲットウインドウ内でrte4win32の内部コマンドのトレースコマンドを用いる方法の2通りがあります。これらを同時に使用することはできませんので、Multiのトレース機能(TimeMachine)を使う場合は、内部コマンドでトレース関連のコマンドは使用しないでください。また、逆にターゲットウインドウ内で内部コマンドでトレースを使用する場合は、TimeMachineを無効に設定し、初期設定が残っている場合は、再設定し直してからご使用ください。

※2010年6月14日現在、TimeMachineは未対応です。

実行中のトレース表示

内蔵ROMの空間で実行するプログラムは、実行中でもトレース表示が可能です。これは、デバッガからプログラムをダウンロードする時点でPC上のメモリにプログラムの内容を写像(キャッシング)しているためです。但し、デバッガが実行中のトレース表示を禁止している場合はできません。

キャッシングは、内部コマンドのifromenvコマンドで設定します。

トレースの中止

デュアルコアのCPUをICEしている場合、どちらかのコアでトレースを中止すると、両コアのトレースが中止されます。また、どちらかのコアでトレース結果の表示を行った場合も同様に両コアのトレースを中止します。

内蔵フラッシュROMへの書き込み

デュアルコアのCPUでは、内蔵フラッシュROMへの書き込みは両コアがブレーク中でなければなりません。自動書き込みの場合は、一度両コアがブレークし、その後どちらかのコアの実行開始時に自動で書き込みます。

同時ブレーク/実行、同期ブレーク

MULTIデバッガのタスクマネージャを使用して、デュアルコアの両コアを同時にブレークしたり、実行したりすることができますが、これは、ソフト処理によるものですので、両コアの間でソフト処理分の時間差が生じます。

しかし、ブレークは、autobreak機能(デフォルトでON)により、両コアのハードウェアによる同期ブレーク機能を使うことで、両コアの間での時間差はほとんどありません。

特定レジスタのシーケンスアクセス

ICEはブレーク中のデバッガからのアクセスが特定レジスタのアクセスシーケンスを妨害しない機能を有効にしていますが、デュアルコアの場合にこの機能が有効になるのは、両コア共にブレークしたときです。どちらかのコアが実行中のときは、この機能は無効ですので、特定レジスタのシーケンス中に片方のコアのみがブレークし、デバッガから操作をした場合は、アクセス中のシーケンスを妨害する可能性があります。

ブレーク中のタイマ動作

ICEはブレーク時にタイマ、シリアルインターフェース、A/Dコンバータのカウント動作を停止する機能を持ち、内部コマンドのenvコマンドのsvstopパラメータで設定できます。デフォルトではOFFになっていますので、必要に応じて設定してください。

➔ デュアルコアでこの機能を有効にした場合は、どちらかのコアがブレークするとカウント動作が停止しますので、別々に実行・ブレークの操作している場合は注意してください。

レジスタ

次のレジスタの値は変更できません。デバッガでは見かけ上変更できるものがありますが、ICEで専有するレジスタですので、CPUには反映されません。

DBPC, DBPSW, DIR, BPAV, BPAM, BPDV, BPDM, DBWR

INITコマンド

内部コマンドのINITコマンドをデュアルコアで使用した場合は、両コアが初期化されます。

HALT命令

HALT命令でブレークした場合、ブレーク時のアドレスは、HALT命令の次の命令の先頭アドレスになります。

その他

製品にリリース・ノート等が添付されている場合は必ずそれらも参照ください。

付録. A トレース機能の詳細

内部コマンドを使用したリアルタイムトレース機能について説明します。

トレースの概要

リアルタイムトレースは、CPUおよびDMAから出力された実行内容（トレースデータ）を、実行ごとにICE内のトレースバッファに書き込みます。この内容は、“trace”コマンドで見ることができます。

トレースデータの取り込みは、トレースモード、トレース開始条件、トリガ条件、セクション条件、クォリファイ条件などの設定によって指定できます。トレースデータ取り込みの流れについては、図 1、図 2を参照してください。

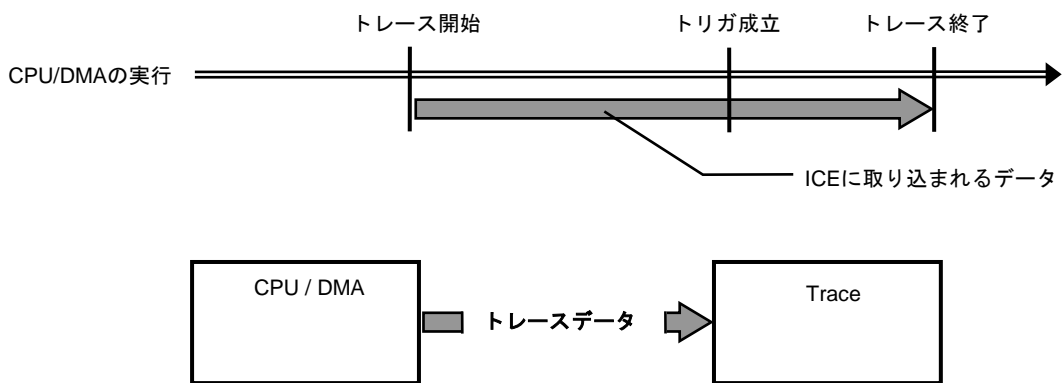


図 1 トレースデータ取り込みの流れ

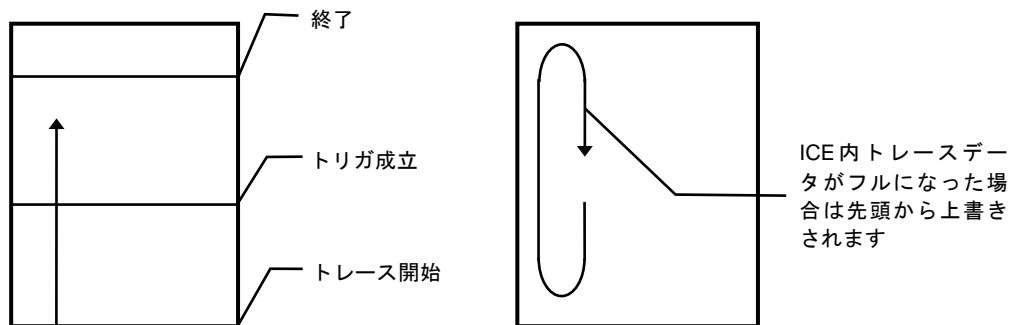
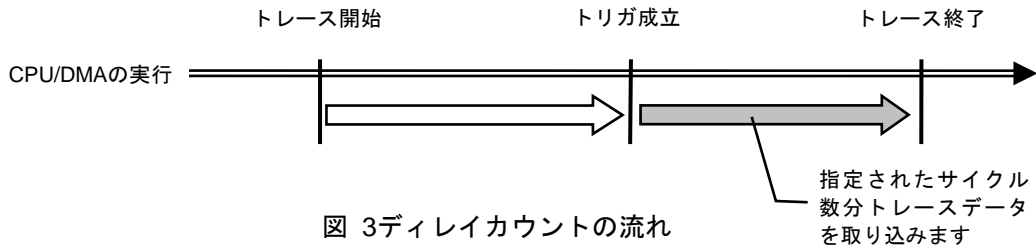


図 2 ICE内のトレースデータ

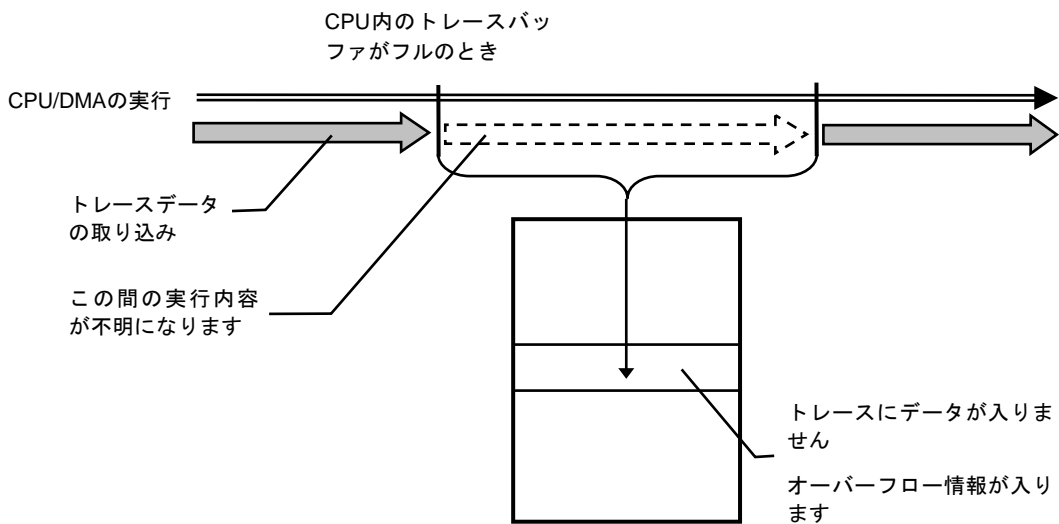
ディレイカウント

ディレイカウントは、トリガ成立後に取り込むサイクル数です(図 3)。サイクル数は、CPUの実行内容により異なります。1サイクルが1実行単位ではありません。



トレースの実行モード

リアルタイム・モードは、CPU/DMAの実行を優先してトレースデータを取り込むモードです。CPU内のトレースバッファ(FIFO)がフルになった場合、トレースデータの取りこぼしが発生することがあります(図 4)。



非リアルタイム・モードは、トレースデータの取りこぼしがないようにするモードです。このモードでは、CPU内のトレースバッファ (FIFO) がフルになった場合、CPU/DMAの実行を一時停止し、その後自動的に再開します (図 5)。

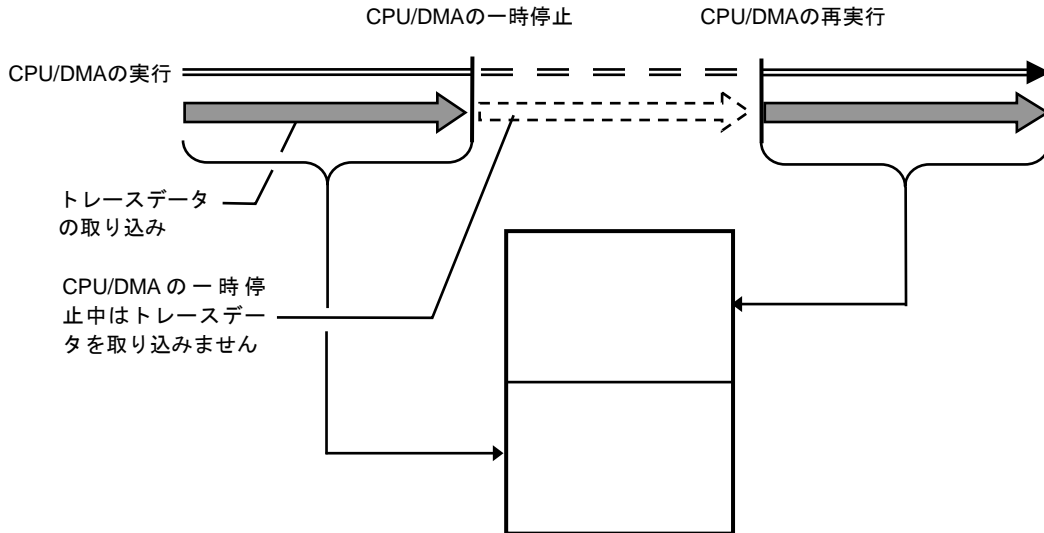


図 5 非リアルタイム・モード

サブスイッチ、セクション、クオリファイ

サブスイッチは、セクション条件の成立状態とクオリファイ条件の成立状態のandまたは、orの組み合わせによる状態をいい (tenv [subor|suband]、dmatenv [subor|suband]により設定)、成立の状態をon、不成立の状態をoffと定め、このon/offそれぞれの状態で、トレースに取り込むサイクルを指定することができます (sswon/dmasswon, sswoff/dmasswoffコマンド)。通常、sswon/dmasswonに取り込みたいサイクルを指定し、sswoff/dmasswoffに何も取り込まない設定をすることで、このサブスイッチのon/offの状態がトレースの開始と停止に対応します。(sswon/dmasswon, sswoff/dmasswoffコマンドの初期値はこのようになっています。これ以降この設定になっていることを前提で説明します。)

セクションは、tsp1/2, dmatsp1/2コマンドと evt secon/secoff, dmaevt secon/secoffの条件で指定できます。tsp1, dmatsp1, seconがセクションの成立条件(on)、tsp2, dmatsp2, secoffが非成立条件(off)となります。

クオリファイ条件は、evt/dmaevtコマンドでqualifyに指定したイベントの条件成立がそのままクオリファイの成立となります。

サブスイッチに用いるそれらの条件はsswon/dmasswon, sswoff/dmasswoffコマンドで選択します。

トレースの開始

トレースの取り込みを開始するには、強制的に開始する方法 (tron force) と、セクションとクオリファイの設定に基づく、サブスイッチの状態で行う方法があります。(図 6)

サブスイッチによる取り込み条件の設定は、sswon/dmasswon, sswoff/dmasswoffで指定します。通常、sswon/dmasswonに取り込みたいサイクルを指定し、sswoff/dmasswoffに何も取り込まない設定をすることで、サブスイッチがonの状態ではトレースを取り込み、サブスイッチがoffの状態ではトレースの取り込みを停止することができます。

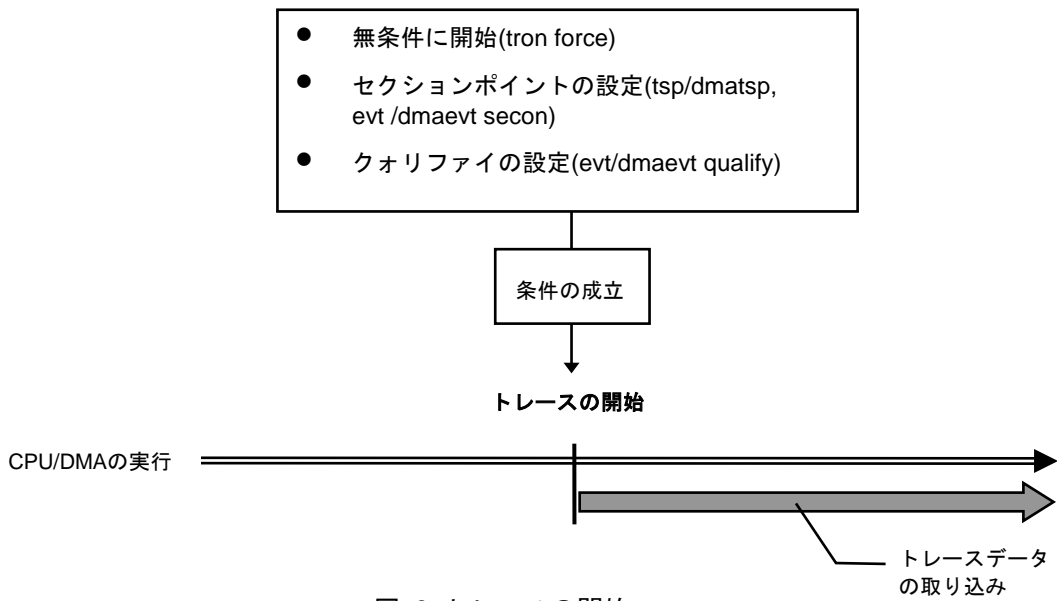


図 6 トレースの開始

トリガ条件

ディレイカウンタの起点となる条件です(図 7)。トリガ条件を設定することにより、条件前後の実行内容を見ることができます。

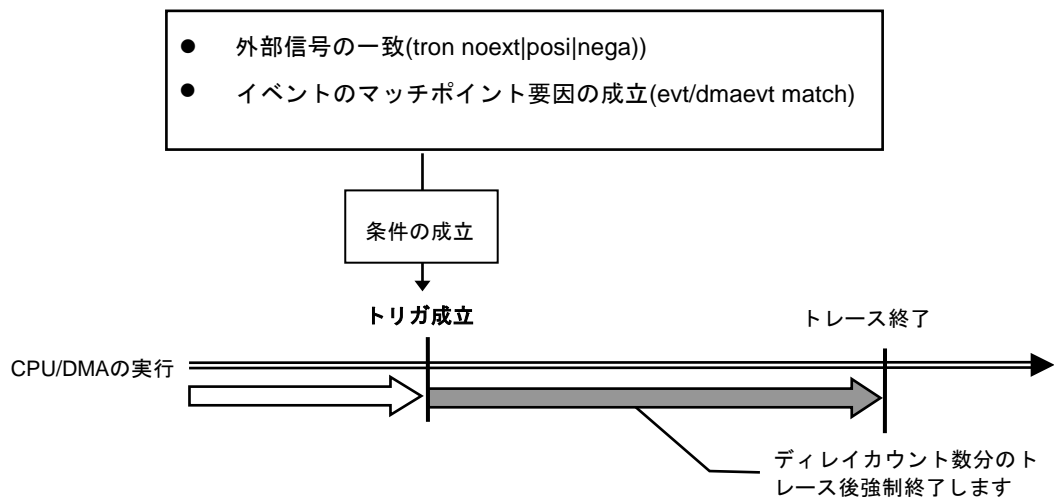


図 7 トリガ条件

トレースの停止

トレースの取り込みを停止するには、セクションとクオリファイの設定に基づく、サブスイッチの状態で行います。(図 8)

サブスイッチによる取り込み条件の設定は、sswon/dmasswon, sswoff/dmasswoffで指定します。通常、sswon/dmasswonに取り込みたいサイクルを指定し、sswoff/dmasswoffに何も取り込まない設定をすることで、サブスイッチがonの状態ではトレースを取り込み、サブスイッチがoffの状態ではトレースの取り込みを停止することができます。

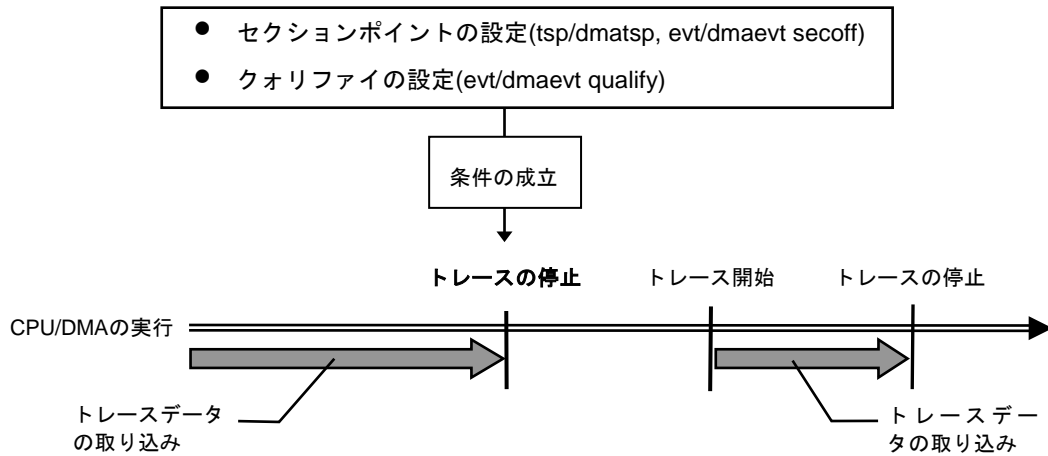


図 8 トレースの停止

トレースの終了

トレースの終了時は、以降のトレースデータの取り込みをしません。停止条件とは違い、再度トレースを開始することはありません(図 9)。

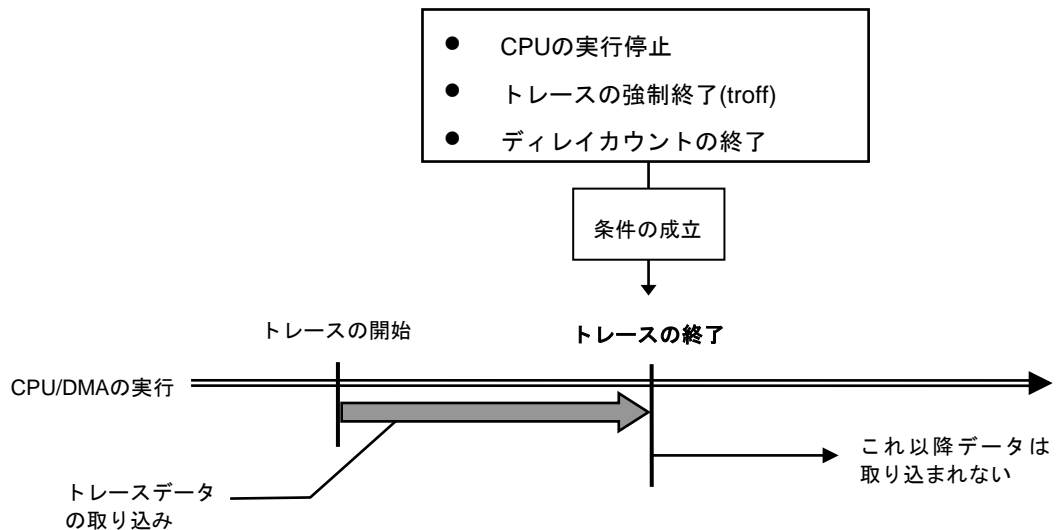


図 9 トレースの終了

強制ディレイモード

強制ディレイモードは、トレース開始後、指定されたディレイカウント(サイクル数)分取り込んだ時点で強制的にトレースを終了します。このモード中はトリガ条件を無視します(図10)。

この場合のトレース開始は、CPU/DMAの実行開始です。

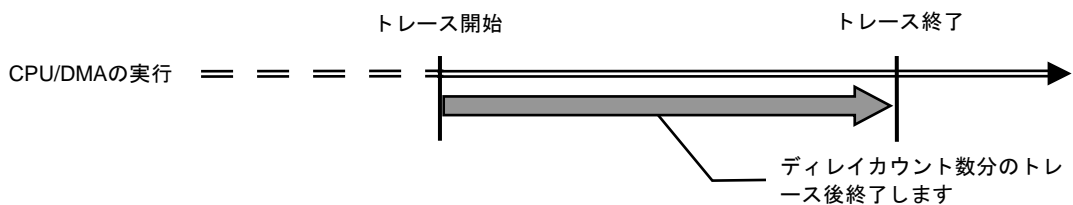


図 10 強制ディレイモード