|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 通知先 | RH850G3/G4コア用 ICE  EXECインターフェース関数仕様書 | LLWEB- 10008657 | | | 1/450 | |
| 2019年 5月 13日 | | | | |
| ルネサス エレクトロニクス株式会社  車載情報プロジェクトマネジメント統括部  車載情報プロジェクト計画部  第三課 | | | | |
| 承 認 | 査 閲 | 作 成 | |
|  |  |  | |

本仕様書は、RH850G3/RH850G4コア用EXECインターフェース仕様を定めたものである。

DCU(Debug Control Unit)を内蔵したRH850G3/RH850G4コア用デバイスとE1/E20/E2/IE850A/IE850をコントロールするソフトウェア(以降EXEC)を呼び出す関数について記述してある。

関連文書

・V850 E2Rコア用EXECインターフェース関数仕様書 IDF-04-026401-13

・V850E3v5アーキテクチャ仕様書 IDS-5392-08

・統合コアG3M CPUサブシステム仕様概要書 -----------

・G3MSS HW 仕様書(for PFC1A)(Ver0.03) -----------

・G3世代JTAG通信ユニット(JCU)機能仕様書 LLWEB-00005329-05

・G3世代少ピン・デバッグ・ユニット(LDU)機能仕様書 LLWEB-00006125-05

・G3/G4世代メモリ・アクセス・ユニット(MAU)機能仕様書 LLWEB-00006144-06

・G3世代RUNコントロール・ユニット(RCU)機能仕様書 LLWEB-00006163-06

・G3世代キャリブレーション機能ユニット(CFU)機能仕様書 LLWEB-00006255-01

・G3世代ソフト・トレース・コントロール・ユニット(TCU\_SFT)機能仕様書 LLWEB-00006355-05

・G3世代トレース・コントロール・ユニット(TCU)機能仕様書 LLWEB-00006816-05

・G3世代CPU・インタフェース・ユニット(CIU)機能仕様書 LLWEB-00006818-05

・G3世代トリガ・イベント・ユニット(TEU)機能仕様書 LLWEB-00010730-05

・G3世代時間／パフォーマンス計測・ユニット(TMU\_TCK/TMU\_CLK)機能仕様書 LLWEB-00010732-05

・RV40F FPSYS/FACI Target Specification Ver.05.07.E WEB-00496435

・System Specification for Standard Boot Firmware

on the RV40F Flash based Microcontrollers IDS-5582

・E1用アイソレータ(RH850\_RL78対応)R0E000010ACB20機能設計書 SPF-R0E000010ACB20-C

・サポートマイコンの各目標仕様書 -----------

・RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書 LLWEB-10008627

・G4世代少ピン・デバッグ・ユニット(LDU)機能仕様書 LLWEB-????????-00(未採番)

・G4世代トリガ・イベント・ユニット(TEU)機能仕様書 LLWEB-????????-00(未採番)

・G4世代トレース・コントロール・ユニット(TCU)機能仕様書 LLWEB-????????-00(未採番)

・G4世代RUNコントロール・ユニット(RCU)機能仕様書 LLWEB-????????-00(未採番)

・G4世代時間／パフォーマンス計測・ユニット(TMU\_TCK/TMU\_CLK)機能仕様書 LLWEB-????????-00(未採番)

・E2エミュレータ 拡張機能部 インタフェース関数仕様書 IDF-16-014358

・E2エミュレータ 拡張機能部 F/W機能仕様書 IDF-16-014359

・RH850G4対応 デバッガシステム仕様書 未採番

　・IE850A FPGA詳細仕様書 IDF-17-004868

目 次

[1. 概要 7](#_Toc4141523)

[1.1 位置付け 11](#_Toc4141524)

[1.2 本仕様書で使用している記述説明と命名規則 15](#_Toc4141525)

[1.2.1 記述説明 15](#_Toc4141526)

[1.2.2 命名規則 16](#_Toc4141527)

[1.3 ヘッダファイル 17](#_Toc4141528)

[1.4 インターフェースを提供するプログラムについて 18](#_Toc4141529)

[1.4.1 ソフトウェアの構成および動作環境について(E1/E20) 18](#_Toc4141530)

[1.4.2 ソフトウェアの構成および動作環境について(E2/IE850A) 19](#_Toc4141531)

[1.4.3 ソフトウェアの構成および動作環境について(IE850) 20](#_Toc4141532)

[1.4.4 動作環境についての注意点 21](#_Toc4141533)

[1.4.5 プログラムの構成について 21](#_Toc4141534)

[1.4.6 EXRH850Gx.DLLファイルのプロパティについて 22](#_Toc4141535)

[1.4.7 提供ファイルについて 23](#_Toc4141536)

[2. インターフェース仕様 24](#_Toc4141537)

[2.1 インターフェース関数 24](#_Toc4141538)

[2.1.1 インターフェース関数の分類 24](#_Toc4141539)

[2.1.2 インターフェース関数一覧 26](#_Toc4141540)

[2.2 起動と終了 30](#_Toc4141541)

[2.2.1 起動 30](#_Toc4141542)

[2.2.2 終了 38](#_Toc4141543)

[2.3 CPUリセットが発生する関数 39](#_Toc4141544)

[2.4 低消費電力モード時の動作仕様 39](#_Toc4141545)

[2.5 ターゲット電源に関する動作仕様 41](#_Toc4141546)

[2.6 仮想化支援機能に関する動作仕様 42](#_Toc4141547)

[2.7 非同期デバッグに関する動作仕様 44](#_Toc4141548)

[2.8 E2エミュレータ拡張機能に関する動作仕様 48](#_Toc4141549)

[2.9 外部ソフトトレース単線出力機能に関する動作仕様 50](#_Toc4141550)

[2.10 CANブートデバッグに関する動作仕様 54](#_Toc4141551)

[2.11 初期停止・スタンバイモードデバッグ環境に関する動作仕様 55](#_Toc4141552)

[3. EXECインターフェース関数仕様 62](#_Toc4141553)

[3.1 記述規定 62](#_Toc4141554)

[3.1.1 名称で示される入力範囲規定 65](#_Toc4141555)

[3.2 関数詳細説明 66](#_Toc4141556)

[3.2.1 起動、終了、リセット 67](#_Toc4141557)

[ex\_exit 68](#_Toc4141558)

[ex\_initexec 70](#_Toc4141559)

[ex\_reset 74](#_Toc4141560)

[ex\_setemulator 76](#_Toc4141561)

[ex\_getemulist 80](#_Toc4141562)

[ex\_extpower 82](#_Toc4141563)

[3.2.2 ハードウェア条件設定 85](#_Toc4141564)

[ex\_getenv 86](#_Toc4141565)

[ex\_rh\_getenv 88](#_Toc4141566)

[ex\_getstatus 108](#_Toc4141567)

[ex\_setcallback\_sts 116](#_Toc4141568)

[ex\_boardinfo 117](#_Toc4141569)

[ex\_boardinfo2 118](#_Toc4141570)

[ex\_setenv 128](#_Toc4141571)

[ex\_initimem 130](#_Toc4141572)

[ex\_clock 131](#_Toc4141573)

[ex\_mapping 132](#_Toc4141574)

[ex\_pincont 133](#_Toc4141575)

[ex\_m\_montimeout 135](#_Toc4141576)

[ex\_socunitinfo 136](#_Toc4141577)

[ex\_socunitinfo2 137](#_Toc4141578)

[3.2.3 メモリ操作 192](#_Toc4141579)

[ex\_memread 193](#_Toc4141580)

[ex\_memwrite 199](#_Toc4141581)

[ex\_memfill 210](#_Toc4141582)

[ex\_memcopy 214](#_Toc4141583)

[ex\_memsearch 216](#_Toc4141584)

[ex\_memcompare 217](#_Toc4141585)

[3.2.4 レジスタ操作 218](#_Toc4141586)

[ex\_regread 219](#_Toc4141587)

[ex\_regwrite 222](#_Toc4141588)

[3.2.5 イベント条件 224](#_Toc4141589)

[ex\_evncond 225](#_Toc4141590)

[ex\_rh\_evncond 226](#_Toc4141591)

[ex\_evnintgcond 240](#_Toc4141592)

[ex\_rh\_evnintgcond 241](#_Toc4141593)

[3.2.6 実行 246](#_Toc4141594)

[ex\_run 247](#_Toc4141595)

[ex\_resetrun 250](#_Toc4141596)

[ex\_step 252](#_Toc4141597)

[ex\_stop 255](#_Toc4141598)

[ex\_hwbrkevent 256](#_Toc4141599)

[ex\_swbrkcond 259](#_Toc4141600)

[3.2.7 アナライザ 263](#_Toc4141601)

[(1) トレース関数 264](#_Toc4141602)

[ex\_trcsw 270](#_Toc4141603)

[ex\_trcrun 272](#_Toc4141604)

[ex\_trcmode 274](#_Toc4141605)

[ex\_trcclear 278](#_Toc4141606)

[ex\_trcevent 280](#_Toc4141607)

[ex\_trcsearch 283](#_Toc4141608)

[ex\_trcseek 284](#_Toc4141609)

[ex\_trcdelay 285](#_Toc4141610)

[ex\_trcread 287](#_Toc4141611)

[ex\_m\_trceventsel2 288](#_Toc4141612)

[ex\_rh\_trceventsel2 289](#_Toc4141613)

[ex\_m\_trcbufcont 296](#_Toc4141614)

[ex\_rh\_trcbufcont 297](#_Toc4141615)

[ex\_trcrawmeminfo 301](#_Toc4141616)

[ex\_trcrawmemread 303](#_Toc4141617)

[ex\_trchwtune 306](#_Toc4141618)

[ex\_rh\_trchwevncopeinfo 307](#_Toc4141619)

[ex\_rh\_trctune 309](#_Toc4141620)

[ex\_rh\_trcfilter 323](#_Toc4141621)

[(2) 外部ソフトトレース単線出力関数 325](#_Toc4141622)

[ex\_rh\_sfttrcsw 326](#_Toc4141623)

[ex\_rh\_sfttrcmode 328](#_Toc4141624)

[ex\_rh\_sfttrcevent 331](#_Toc4141625)

[ex\_rh\_sfttrcfilter 332](#_Toc4141626)

[ex\_rh\_sfttrctune 338](#_Toc4141627)

[ex\_rh\_sfttrcrun 343](#_Toc4141628)

[ex\_rh\_sfttrcrawmemmode 345](#_Toc4141629)

[ex\_rh\_sfttrcrawmeminfo 347](#_Toc4141630)

[ex\_rh\_sfttrcrawmemread 349](#_Toc4141631)

[(3) カバレッジ関数 352](#_Toc4141632)

[ex\_covmapping 353](#_Toc4141633)

[ex\_covmapping2 354](#_Toc4141634)

[ex\_covsw 355](#_Toc4141635)

[ex\_covrun 356](#_Toc4141636)

[ex\_covclear 357](#_Toc4141637)

[ex\_covsearch 358](#_Toc4141638)

[ex\_covread 360](#_Toc4141639)

[ex\_covwrite 362](#_Toc4141640)

[(4) タイマ関数 / パフォーマンス計測関数 363](#_Toc4141641)

[ex\_timesw 366](#_Toc4141642)

[ex\_timerun 368](#_Toc4141643)

[ex\_timeclear 369](#_Toc4141644)

[ex\_timeread 370](#_Toc4141645)

[ex\_timeread2 371](#_Toc4141646)

[ex\_rh\_timeread 375](#_Toc4141647)

[ex\_timeevent2 377](#_Toc4141648)

[ex\_timeevent3 382](#_Toc4141649)

[ex\_rh\_timeevent 384](#_Toc4141650)

[ex\_timemode 389](#_Toc4141651)

[ex\_rh\_performancesw 390](#_Toc4141652)

[ex\_rh\_performanceclear 392](#_Toc4141653)

[ex\_rh\_performanceread 393](#_Toc4141654)

[ex\_rh\_performanceset 395](#_Toc4141655)

[3.2.8 ハードウェア直接操作用関数 401](#_Toc4141656)

[D\_Command 402](#_Toc4141657)

[E\_Command 403](#_Toc4141658)

[ex\_dcuregread 404](#_Toc4141659)

[ex\_dcuregwrite 406](#_Toc4141660)

[ex\_setselid 407](#_Toc4141661)

[ex\_getselid 408](#_Toc4141662)

[ex\_setnid 409](#_Toc4141663)

[ex\_getnid 410](#_Toc4141664)

[3.3 エラーコード 411](#_Toc4141665)

[3.4 レジスタID 413](#_Toc4141666)

[4. 非同期通信インターフェース仕様 414](#_Toc4141667)

[4.1 概要 414](#_Toc4141668)

[4.2 インターフェース仕様 414](#_Toc4141669)

[5. ログ情報の出力仕様 419](#_Toc4141670)

[5.1 概要 419](#_Toc4141671)

[5.2 INIファイル仕様 420](#_Toc4141672)

[5.3 ログ情報の出力切り替え用EXEC I/F仕様 422](#_Toc4141673)

[5.4 ログ情報フォーマット仕様 422](#_Toc4141674)

＜仕様書記述形式の注意事項＞

・余白には文字、図形を問わず、何も記載しないこと

・カタカナは全角に統一すること

・スペースは半角に統一すること

・アルファベットは半角に統一すること

・カラムを区切りたい場合は、表(セル)を使うこと

・奇数個の半角文字使用時にカラムが合わない場合は、半角文字の最後にスペースを追加すること

・WORD機能オートコレクトオプションのオートコレクト機能はオフにして編集すること

# 概要

本仕様書は、IE(インサーキット・エミュレータ)の機能を制御するプログラム(以降EXEC)を呼出すためのインターフェース(以降、EXECインターフェースと呼ぶ)と非同期処理のインターフェースについて記載する。本仕様書では、下記の表にあるG3 V4.04、G4 V1.02にてサポートする機能について記載する。

表 1‑1 RH850 G3世代MCUのサポート機能一覧

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | 機能 | | | | | | Phase1.0 | Phase2.0 | Phase2.1 | Phase2.2 | Phase2.3 | Phase2.4 | Phase2.5 | Phase2.6 | V3.0 | V3.1 | V4.0 | V4.01 | V4.02 | V4.03 | 4.04 | EML種 | 備考 |
| 1 | サポート  マイコン | E1x-FCC(PFC1A) | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 2 | F1L(-GW) | | | | | △※ | ○ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | ※V1のみサポート |
| 3 | R1L | | | | | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 4 | E1x-FCC1(PFC1B)/E1x(160/320MHz) | | | | | - | ○ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | Phase2.1　320MHzPod追加 |
| 5 | P1M | | | | | - | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 6 | C1H/C1M | | | | | - | - | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 7 | F1H(-GW)/F1M | | | | | - | - | - | ○ | ←※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | ※ICUM-ON、C&Rのサポート |
| 8 | F1K | | | | | - | - | - | - | - | △※1 | ○ | ← | ←※2 | ← | ← | ← | ← | ← | ← | OCD | ※1　WSのみ  ※2　1stCut未サポート |
| 9 | F1KM-S1、F1KM-S4 | | | | | - | - | - | - | - | - | - | - | - | - | ○ | ← | ← | ← | ← | OCD | － |
| 10 | F1KH-D8 | | | | | - | - | - | - | - | - | - | - | - | - | - | ○ | ← | ← | ← | OCD | － |
| 11 | CC-Cube(D5ED, D4, D3) | | | | | - | - | - | △※1 | ←※2 | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | ※1　WSのみ  ※2　ICUM-ON、C&Rのサポート |
| 12 | P1x-C(一般品) | | | | | - | - | - | - | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | ICUM-ON、C&Rのサポート含む |
| 13 | D1x | | | | | - | - | - | ○※1 | ← | ←※2 | ← | ← | ← | ←※3 | ← | ← | ← | ← | ← | ALL | ※1　D1M2 PODのみ  ※2　D1M1 POD追加  ※3 D1x SuperEVA搭載POD追加(WSのみ) |
| 14 | E1x-FCC2/E1M-S2/E1M-A | | | | | - | - | - | - | △※1 | ○ | ← | ← | ← | ← | ← | ← | ←※2 | ← | ← | ALL | ※1　WSのみ  ※2　E1M-A追加 |
| 15 | V1R-M | | | | | - | - | - | - | - | - | △※ | - | △※ | ← | ○ | ← | ← | ← | ← | OCD | ※WSのみ |
| 16 | C1M-A | | | | | - | - | - | - | - | - | △※ | - | ○ | ← | ← | ← | ← | ← | ← | OCD | ※WSのみ |
| 17 | P1M-E | | | | | - | - | - | - | - | - | △※ | - | ○ | ← | ← | ← | ← | ← | ← | OCD | ※WSのみ |
| 18 | R-Car V3H(ICUMXA) | | | | | - | - | - | - | - | - | - | - | - | - | - | - | ○ | ← | ← | E2 |  |
| 19 | デバイスファイル対応 | | | | | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 20 | サポート  エミュレータ | E1/E20 | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 21 | E2 | | | | | - | - | - | - | - | - | - | ○※1 | ← | ←※2 | ← | ← | ← | ← | ← | OCD | ※1サポートマイコンはD1x限定  ※2全サポートマイコンに対応 |
| 22 | IE850 | | | | | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | FULL | － |
| 23 | 接続・切断 | MCUとの通信方式(LPD) | | 1pin | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | 製品仕様により未サポートあり |
| 24 | 4pin | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 25 | Hot Plug-in | | | | | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | ホットプラグイン用アダプタが必要 |
| 26 | 複数台接続 | | | | | － | ○ | ← | ←※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※OCD/IE850のそれぞれの最大接続台数を4台とする。 |
| 27 | 指定したシリアル番号を持つ  エミュレータへの接続 | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | － |
| 28 | 指定したニックネームを持つ  エミュレータへの接続 | | | | | － | － | － | ○ | ← | ← | ← | ←※ | ← | ← | ← | ← | ← | ← | ← | FULL | ユーティリティツールによりニックネームをエミュレータに書き込む。  ※E2エミュレータでもサポート |
| 29 | 起動時のモード変更  (E1x系/C1x系のみ) | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | シリアルプログラミングモードの場合はユーザブートモードに変更して起動する |
| 30 | 電源供給機能 | | | | | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | 使用目的はアイソレータ専用とする。 |
| 31 | リニア電源供給機能  接続後の供給電圧変更 | | | | | － | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | OCD | E1/E20エミュレータではサポートしない |
| 32 | FCCをデバッグチップとして  デバッグする | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | OCD | 新接続形態 |
| 33 | 接続時の自走時間短縮 | | | | | － | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | LPD 4pin接続時のみ(LPD 1pinは除く) |
| 34 | 切断時のリセット端子制御 | | | | | － | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | E2 | － |
| 35 | エミュレータリスト取得 | | | | | － | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | OCD | － |
| 36 | ダウンロード・  アップロード機能 | ROM(Flash)、RAM | | | | | ○ | ← | ← | ← | ←※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※フラッシュ書き換え時にDMAを一時的に停止する。 |
| 37 | Data Flash | | | | | ○ | ← | ← | ←※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※データフラッシュ仕様に合わせた書き込みサイズ対応およびブランクチェックコマンドのサポート |
| 38 | 外部RAMへのダウンロード | | | | | － | － | － | △※ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※D1xのみサポート |
| 39 | 外部Flashダウンロード | | | | | － | － | － | － | － | － | － | － | － | － | － | ← | ← | ← | ← | ALL | － |
| 40 | オプションバイト書き換え | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 41 | セキュリティフラグエミュレーション | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 42 | LockBit対応 | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 43 | OCD ID対応 | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 44 | IDTAGヘキサフォーマット対応 | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | 案件中止のためサポート予定なし。 |
| 45 | ワークRAM領域の設定/参照 | | | | | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 46 | メモリアクセス | 設定・参照 | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 47 | Fill | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 48 | メモリコピー | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 49 | 外部空間へのアクセス | | | | | － | － | － | △※ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※D1xのみサポート |
| 50 | レジスタ・SFR機能 | 設定・参照 | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 51 | プログラム 実行機能 | リアルタイム実行 | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 52 | ステップ実行 | | | | | ○※1 | ○※2 | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※1　カレントコアのみ  ※2　全コア |
| 53 | イベント機能 | 実行検出(実行PC) | | | | 8本/コア | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 54 | CPUアクセス検出 | | | | 8本/コア | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 55 | DMAバス検出 | | | | 4本 | － | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 56 | ブレークアクション | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | シーケンシャル・累積AND・パスカウント、範囲内・範囲外、ディレイブレーク |
| 57 | トレース  アクション | | セクション | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 58 | クオリファイ | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | 範囲内・範囲外 |
| 59 | トレース終了 | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | シーケンシャル・累積AND・パスカウント、ディレイ、範囲内・範囲外 |
| 60 | 区間計測アクション | | | | | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | 時間測定機能-区間測定 |
| 61 | ブレーク機能 | PC実行(ハード) | | 12本/コア※ | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※4本はCPUアクセスと共通 |
| 62 | CPUアクセス  (ハード) | | 4本/コア | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 63 | ソフトブレーク　(合計2000本) | | 内蔵ROM | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 64 | ERAM | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | サポート予定なし。ERAMを操作する  ユーザプログラムのデバッグは可能。 |
| 65 | 内蔵RAM | | | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 66 | 外部ROM | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 67 | 外部RAM | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 68 | 強制ブレーク | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 69 | トレース機能 | 全トレース | リアルタイム | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 70 | ノンリアルタイム | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 71 | トレース  アクション | フル停止 | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 72 | フルブレーク | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 73 | ディレイ停止 | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 74 | ディレイブレーク | | | | - | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | FULL | IE850外部トレース選択時のみ使用可能 |
| 75 | トレース対象 | CPU | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 76 | DMA | | | | － | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 77 | ソフトトレース | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 78 | タイムスタンプ | 積算 | | | | － | ○ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | F1L(v1)使用不可。 |
| 79 | トレース方式 | MCU内蔵メモリ | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 80 | 専用トレース端子 | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | FULL | MCU内蔵メモリと同時使用不可。 長時間トレースオプションのサポート。 フルブレーク/フルストップ/ディレイブレーク/ディレイストップのサポート |
| 81 | LPD端子 | | | | － | － | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | E2 | － |
| 82 | HDD出力 | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | FULL | ハードディスクドライブへのトレース・データを直接書き込むことで、テラバイトオーダートレース・データを取得できる機能 |
| 83 | 時間測定機能 | Run-Break測定 | | | 1本/コア | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | TMU\_TCK(LPDクロック) |
| 84 | 区間測定：TCK/CPUクロック  累積、最大、  最小、回数 | | | 7本/コア | | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | TMU\_TCK(LPDクロック)x3本 TMU\_CLK(CPUクロック)x4本 |
| 85 | パフォーマンス測定 | | | 4本/コア | | － | － | － | － | － | ○※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※一部未サポートの計測項目あり (TMU\_CLKとの排他利用) |
| 86 | 実行中のメモリR/W操作(RRM/DMM) | | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | MAUアクセス可能領域のみRRM/DMMをサポート。CPUコアレジスタは不可、IORレジスタはアクセス可能。ただし、アクセス不可のIORに対しては、EXECI/F呼び出し元モジュールにてDFファイルの情報より判断しアクセスしないこと。 |
| 87 | カバレッジ機能 | | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | FULL | － |
| 88 | 端子マスク機能 | | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 89 | ペリフェラルブレーク機能 | | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 90 | デバッグコンソール機能 | | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 91 | エネルギープロファイル機能 | | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | FULL | 消費電流の変動をプログラムの動きとあわせてトレースする。消費電流のピーク等の分析に役立てることを想定した機能。 |
| 92 | エミュレータ接続検出用デバッグ情報出力機能 | | | | | | － | － | － | － | － | － | － | － | － | ○ | ← | ← | ← | ← | ← | FULL | ※一部未サポートのデバイス品種あり |
| 93 | マルチコア対応 | 同期実行・同期ブレーク | | | | | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | － |
| 94 | 個別実行・個別ブレーク | | | | | － | － | － | △※ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※WS対応のみ |
| 95 | 複数指定コア実行・ブレーク(デバッグスコープ) | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ALL | － |
| 96 | 初期停止エミュレーション | | | | | － | － | － | △※ | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※WS対応のみ |
| 97 | 仮想マシン対応(G3世代) | 実行 | | | ステータス | | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | NM/VMの実行ステータス取得 |
| 98 | ブレーク | | | NM/VM指定 | | － | － | △※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ※NMとVMの選択は不可。  NMとVMの両方を有効指定として動作する |
| 99 | トレース | | | NM/VM指定 | | － | － | △※ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL |
| 100 | レジスタ | | | NM/VM対応 | | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | NM/VM両コンテキストの個別アクセス |
| 101 | パフォーマンス | | | NM/VM対応 | | － | － | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | NM/VM両コンテキスト選択 |
| 102 | PRDNAME,PRDSEL対応 | | | | | | － | ○ | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ALL | ICEチップ時に対象デバイス情報に変更する |
| 103 | メインコアデバッグ | | | | | | － | － | － | － | － | － | － | － | － | － | － | ○※ | ← | ← | ← | ALL | ※F1KH-D8、F1KM-S4のみ |
| 104 | 初期停止・スタンバイモードデバッグ環境 | | | | | | － | － | － | － | － | － | － | － | － | － | － | － | － | － | ○ | All | ※F1KH-D8、F1KM-S4のみ |

○：サポート予定(V4.03以前の記載ではサポート済みを意味する)

－：サポートなし

←：継続サポート

△：機能制限あり

ALL：E1/E20/E2/IE850エミュレータのサポート

OCD：E1/E20/E2エミュレータのみのサポート

FULL：IE850エミュレータのみのサポート

表 1‑2 RH850 G4世代MCUのサポート機能一覧

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | 機能 | | | | | | 2017/1 WS1 | 2017/3 WS2 | 2017/7 WS3 | 2017/12 WS4 | V1.00 | V1.01 | V1.02 | EML種 | 備考 |
| 1 | サポート マイコン | RH850 E2x-FCC1 | | | | | △※1 | ← | ← | ← | ○ | ← | ← | ALL | ※1　WSのみ |
| 2 | RH850 E2M | | | | | － | － | － | △※1 | ○ | ← | ← | ALL | ※1　WSのみ |
| 3 | RH850 E2x-FCC2 | | | | | － | － | － | － | ○ | ← | ← | ALL | － |
| 4 | RH850 E2UH/E2H/E2GUH/E2GH | | | | | － | － | － | － | － | － | ○ | ALL | － |
| 5 | RH850 U2A-EVA | | | | | － | － | － | － | － | △  ※1 | ○ | ALL | ※1　WSのみ |
| 6 | サポートエミュレータ | E2 | | | | | ○ | ← | ← | ← | ← | ← | ← | E2 | － |
| 7 | IE850A | | | | | － | － | － | － | － | ○ | ← | IE850A | － |
| 8 | 接続・切断 | 端子制御によるLPD通信への切り替え | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | (G4コア新機能) ライタモードに入らず、起動中だけLPD通信モードへ一時的に変更する |
| 9 | MCUとの通信方式 | | | LPD 1pin | | － | － | － | － | － | － | － | ALL | G4は現状LPD 1pinなし |
| 10 | LPD 4pin | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 11 | Hot Plug-in | | | | | ○ | ← | ← | ← | ← | ← | ← | E2 | Hot Plug-inでは自走停止を抑止する |
| 12 | 複数台接続 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | ※最大接続台数を4台とする。 |
| 13 | 指定したシリアル番号を持つ エミュレータへの接続 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 14 | 指定したニックネームを持つ エミュレータへの接続 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 15 | 起動時のモード変更 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | エミュレータ接続時はSTMSELのシリアルフラッシュモードを一時的に無効化 |
| 16 | リニア電源供給機能 接続後の供給電圧変更 | | | | | ○ | ← | ← | ← | ← | ← | ← | E2 | － |
| 17 | ID認証 | | | | | ○ | ← | ←  ※1 | ← | ← | ← | ← | ALL | OCD-ID、Customer ID、DataFlash IDの3種をサポート  ※1　Serial Programmer ID、RHSIF ID、C-TEST IDの3種もサポート |
| 18 | 通常接続時の自走停止 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | (G4コア新機能) LPD 4pin接続時のみサポート 通常起動時は無条件で自動停止起動 |
| 19 | SVRパラメータ設定 | | | | | － | － | － | － | － | ○ | ← | ALL | SVRパラメータ設定が必要なデバイスのみ使用 |
| 20 | マップモード設定 | | | | | － | － | － | － | － | ○ | ← | ALL | ダブルマップモード搭載デバイスのみ使用可能 |
| 21 | 切断時のリセット端子制御 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 22 | エミュレータリスト取得 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 23 | ダウンロード・アップロード機能 | ROM(Flash) | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 24 | Data Flash | | | | | ○ | ← | ← | ← | ←※ | ← | ← | ALL | ※拡張Data Flash対応 |
| 25 | 内蔵RAMへのダウンロード | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 26 | 外部RAMへのダウンロード | | | | | － | － | － | － | － | － | － | ALL | 未サポート |
| 27 | 外部Flashダウンロード | | | | | － | － | － | － | － | － | － | ALL | － |
| 28 | オプションバイト書き換え | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | OPBT書き換えに対応 |
| 29 | リセットベクタアドレス書き換え | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | (G4コア新機能) オプションバイトの書き換えによる |
| 30 | セキュリティフラグエミュレーション | | | | | － | － | － | － | － | － | － | ALL | － |
| 31 | LockBit対応 | | | | | － | － | － | － | － | － | － | ALL | － |
| 32 | 起動中のOCD ID書き換え | | | | | － | － | － | － | － | － | － | ALL | － |
| 33 | ワークRAM領域の設定/参照 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 34 | メモリアクセス | 設定・参照 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 35 | Fill | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 36 | メモリコピー | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 37 | 外部空間へのアクセス | | | | | － | － | － | － | － | － | － | ALL | 未サポート |
| 38 | アクセスガードエミュレーション | | | | | － | － | － | － | － | － | － | ALL | (G4コア新機能) ユーザ設定中のアクセスガードをメモリアクセス時に有効/無効選択可能 |
| 39 | レジスタ・  SFR機能 | 設定・参照 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 40 | SFR空間へのMAUアクセス | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | EXECがデバイスファイルから自動判別 |
| 41 | プログラム実行機能 | リアルタイム実行 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 42 | ステップ実行 | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | 同期デバッグ時は全コアステップ実行 |
| 43 | イベント機能 | 実行検出(実行PC) | | | 8本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 44 | CPUアクセス検出 | | | 8本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 45 | スレーブバス検出 | | | LRAM | | － | － | － | ○ | ← | ← | ← | ALL |  |
| 46 | CRAM | | － | － | － | ○ | ←※ | ← | ← | ALL | ※複数クラスタに対応 |
| 47 | DMA | | － | － | － | ○ | ← | ← | ← | ALL |  |
| 48 | ブレークアクション | | | | | ○ | ← | ← | ←  ※ | ← | ← | ← | ALL | シーケンシャル・累積AND・パスカウント、範囲内・範囲外、ディレイブレーク  ※E2x-FCC1(1stCut)では、TEU\_CPUの命令系範囲イベント条件は使用できない。 |
| 49 | トレースアクション | | | セクション | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 50 | クオリファイ | | ○ | ← | ← | ← | ← | ← | ← | ALL | 範囲内・範囲外 |
| 51 | トレース終了 | | ○ | ← | ← | ← | ← | ← | ← | ALL | シーケンシャル・累積AND・パスカウント、ディレイ、範囲内・範囲外 |
| 52 | 区間計測アクション | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | 時間測定機能-区間測定 |
| 53 | ブレーク機能 | PC実行(ハード) | | | 12本/コア※ | | ○ | ← | ← | ← | ← | ← | ← | ALL | ※CPUアクセスと共通 |
| 54 | CPUアクセス(ハード) | | | 4本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | G4世代デバイスではコアイベントのデータ比較が全チャネル使用不可 |
| 55 | ソフトブレーク  (合計2000本) | | | 内蔵ROM | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 56 | ERAM | | － | － | － | － | － | － | － | ALL | サポート予定なし。ERAMを操作するユーザプログラムのデバッグは可能 |
| 57 | 内蔵RAM | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 58 | 外部ROM | | － | － | － | － | － | － | － | ALL | － |
| 59 | 外部RAM | | － | － | － | － | － | － | － | ALL | － |
| 60 | 強制ブレーク | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 61 | トレース機能 | 全トレース | | リアルタイム | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 62 | ノンリアルタイム | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 63 | トレースアクション | | フル停止 | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 64 | フルブレーク | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 65 | ディレイ停止 | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 66 | ディレイブレーク | | | － | － | － | 〇 | ← | ← | ← | IE850A | IE850A外部トレース選択時のみ使用可能 |
| 67 | トレース対象 | CPU | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 68 | スレーブ | | | LRAM | － | － | － | ○ | ← | ← | ← | ALL |  |
| 69 | CRAM | － | － | － | ○ | ←※ | ← | ← | ALL | ※複数クラスタに対応 |
| 70 | DMA | － | － | － | ○ | ← | ← | ← | ALL |  |
| 71 | ソフトトレース命令 | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 72 | タイムスタンプ | | 積算 | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 73 | トレース方式 | | MCU内蔵メモリ | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 74 | Auroraトレース端子 | | | － | － | － | 〇 | ← | ← | ← | IE850A | (G4コア新機能) Aurora I/Fを使用したトレース  MCU内蔵メモリと同時使用不可 |
| 75 | LPD端子 | | | － | － | － | － | ○ | ← | ← | E2 | － |
| 76 | トレースRAMフィルタ | | | | | － | － | － | － | － | － | － | IE850A | (G4コア新機能) LRAM/CRAMに対するトレース出力のフィルタ 大容量トレースとなるIE850A向け機能 |
| 77 | 時間測定 機能 | Run-Break測定 | | | 1本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | TMU\_TCK(LPDクロック) |
| 78 | 区間測定：TCK/CPUクロック 累積、最大、最小、回数 | | | 7本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | TMU\_TCK(LPDクロック)x3本 TMU\_CLK(CPUクロック)x4本 |
| 79 | パフォーマンス測定 | | | 4本/コア | | ○ | ← | ← | ← | ← | ← | ← | ALL | ※一部未サポートの計測項目あり(TMU\_CLKとの排他利用) |
| 80 | 実行中のメモリR/W操作(RRM/DMM) | | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | MAUアクセス可能領域のみRRM/DMMをサポート。RAM、SFR領域はアクセス可能。 但し、アクセス不可のIORに対しては、EXECI/F呼び出し元モジュールにてDFファイルの情報より判断しアクセスしないこと。 |
| 81 | 端子マスク機能 | | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 82 | 周辺ブレーク機能 | 全周辺機能ON/OFF(G3世代相当) | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 83 | モジュール単位でON/OFF(G4新機能) | | | | | － | － | － | － | － | － | － | ALL | (G4コア新機能) ON/OFFをモジュール個別に選択可能 |
| 84 | エミュレータ接続検出用デバッグ情報出力機能 | | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 85 | マルチコア対応 | 同期実行・同期ブレーク | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 86 | 個別実行・個別ブレーク | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 87 | 複数指定コア実行・ブレーク (デバッグスコープ) | | | | | － | － | － | － | － | － | － | ALL | － |
| 88 | 初期停止エミュレーション | | | | | ○ | ← | ← | ← | ← | ← | ← | ALL | － |
| 89 | 初期停止、(Cyclicモード)、状態の検出 | | | | | | － | ○ | ← | ← | ← | ← | ← | ALL | (G4コア新機能) ブレーク後、初期停止検出コアのリレーブレーク要求取り下げる |
| 90 | PRDNAME、PRDSEL対応 | | | | | | － | － | ○ | ← | ← | ← | ← | IE850A | ICEチップ時に対象デバイス情報に変更する |
| 91 | ICUM ONモードのサポート | | | | | | － | ○ | ← | ← | ← | ← | ← | ALL | 詳細はSEC版仕様書に記載 |
| 92 | メインコアデバッグ | | | | | | － | － | － | － | － | ○ | ← | ALL | － |
| 93 | 初期停止・スタンバイモードデバッグ環境 | | | | | | － | － | － | － | － | － | ○ | ALL | － |
| 94 | 仮想マシン対応(G4世代) | | | | | | － | － | － | － | － | － | ○ | ALL | － |

○：サポート予定

－：サポートなし

←：継続サポート

△：機能制限あり

ALL：E2/IE850Aエミュレータのサポート

E2：E2エミュレータのみのサポート

IE850A：IE850Aエミュレータのみのサポート

## 位置付け

＜RH850用EXEC関数基本方針＞

「V850 E2Rコア用EXECインターフェース関数仕様書」に対して関数の使用方法がRH850用となる場合と

RH850用に新設した関数を使用する場合の2種類の対応方法がある。

V850E2用EXEC仕様とRH850用EXEC仕様の関連は下記の表の様になっている。

[EXEC仕様関連表の記載方法]

・RH850用EXEC関数が"-----------------"の場合は本仕様書に記載が存在しない。

・V850E2用EXEC関数が"-----------------"の場合はRH850用に新設した関数となる。

・サポート欄に"×"が付いている関数はEX\_NOSUPRTを返す。

・サポート欄に"--"の場合は未実装(関数が存在しない)となる。

・V850E2用EXEC関数およびRH850用EXEC関数の双方が"-----------------"の場合は本仕様書に記載はないが、現段階でEXECのDLLに関数の実態が存在する。

表 1‑3 EXEC仕様関連表(1/4)

|  |  |  |  |
| --- | --- | --- | --- |
| V850E2用EXEC関数 | RH850用EXEC関数 | 変更項目 | サポート |
| ex\_exit | ex\_exit | 引数の変更なし | ○ |
| ex\_initexec | ex\_initexec | 使用方法変更  ハンドルなし。クロック指定は別関数(ex\_boardinfo2) | ○ |
| ex\_reset | ex\_reset | 引数の変更なし | ○ |
| ex\_setemulator | ex\_setemulator | 引数の変更なし | ○ |
| ----------------- | ex\_getemulist | RH850用EXECで新設 | ○ |
| ----------------- | ex\_extpower | RH850用EXECで新設 | ○ |
| ex\_getenv | ex\_getenv | exec\_info[48]、~~extsupport\_chk~~のみ使用 | ○ |
| ----------------- | ex\_rh\_getenv | RH850用EXEC(Phase2.0)で新設。 | ○ |
| ex\_getstatus | ex\_getstatus | カバレッジの情報は本開発バージョンでは未サポート。  V3.0以降、別関数に変更する予定。 | ○ |
| ----------------- | ex\_setcallback\_sts | RH850用EXECで新設  (非同期通信用コールバック関数の設定) | ○ |
| ex\_boardinfo | ex\_boardinfo | 使用しない(未サポート)  RH850用EXECでは、ex\_boardinfo2のみサポート | × |
| ex\_boardinfo2 | ex\_boardinfo2 | Phase1では、OCDエミュレータのみサポート。  Phase2より、フルエミュレータの定義を追加。 | ○ |
| ex\_setenv | ex\_setenv | pev\_mode、trcmem\_sizeのみサポート。  Phase2より、extsupport\_chkパラメータを無効。 | ○ |
| ex\_initimem | ex\_initimem | 使用しない(未サポート) | × |
| ex\_clock | ex\_clock | 使用しない(未サポート) | × |
| ex\_mapping | ex\_mapping | 使用しない(未サポート) | × |
| ex\_pincont | ex\_pincont | 引数の変更なし | ○ |
| ---------------- | ----------------- | ex\_m\_ieportの記載はV850E2用EXECからない | × |
| ex\_m\_montimeout | ex\_m\_montimeout | 引数の変更なし | ○ |
| ex\_socunitinfo | ex\_socunitinfo | 使用しない(未サポート) | × |
| ex\_socunitinfo2 | ex\_socunitinfo2 | unit\_type追加  Phase2より、下記①～③の情報をex\_rh\_getenvに移動。  ①EX\_UNIT\_SWBRKFUNCTION unit\_info[0] b2  ②EX\_UNIT\_TRCSEL unit\_info[0] b0, b1, b2, b3  ③EX\_UNIT\_TRCSEL unit\_info[2] | ○ |
| ex\_memread | ex\_memread | 使用方法変更。EX\_DATA\_SIZE\_8追加 | ○ |
| ex\_memwrite | ex\_memwrite | 使用方法変更。  modeによりリセットもしくはエラーとなる  (詳細未記載) | ○ |
| ex\_memfill | ex\_memfill | EX\_DATA\_SIZE\_8追加 | ○ |
| ex\_memcopy | ex\_memcopy | EX\_DATA\_SIZE\_8追加 | ○ |

表 1‑4 EXEC仕様関連表(2/4)

|  |  |  |  |
| --- | --- | --- | --- |
| V850E2用EXEC関数 | RH850用EXEC関数 | 変更項目 | サポート |
| ex\_memsearch | ex\_memsearch | 使用しない(未サポート) | × |
| ex\_memcompare | ex\_memcompare | 使用しない(未サポート) | × |
| ex\_regread | ex\_regread | registerID変更。RH850用EXECではEX\_M\_DMM\_REG\_READをサポートしない。 | ○ |
| ex\_regwrite | ex\_regwrite | registerID変更。RH850用EXECではEX\_M\_DMM\_REG\_WRITEをサポートしない。 | ○ |
| ---------------- | ----------------- | ex\_sfrreadの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_sfrwriteの記載はV850E2用EXECからない | × |
| ex\_evncond | ex\_evncond | 使用しない(未サポート)  RH850用EXECは未サポート(ex\_rh\_evncondで対応) | × |
| ---------------- | ex\_rh\_evncond | RH850用EXECで新設  DMAイベント削除。ASIDレジスタ削除。データサイズの  128ビット化。仮想化機能ように引数の追加。イベント  番号体系変更(その他未記載) | ○ |
| ex\_evnintgcond | ex\_evnintgcond | 使用しない(未サポート)  RH850用EXECは未サポート(ex\_rh\_evnintgcondで対応) | × |
| ---------------- | ex\_rh\_evnintgcond | RH850用EXECで新設  パスカウント値の32ビット化。  累積ANDイベントの追加(※シーケンシャルイベントと  排他仕様)  ディレイカウントの追加(※パスカウントと排他仕様)  設定可能イベント条件変更 | ○ |
| ex\_run | ex\_run | 引数の変更なし | ○ |
| ---------------- | ex\_resetrun | RH850用EXECで新設 | ○ |
| ex\_step | ex\_step | 引数の変更なし | ○ |
| ex\_stop | ex\_stop | 引数の変更なし | ○ |
| ex\_hwbrkevent | ex\_hwbrkevent | 引数の変更なし | ○ |
| ex\_swbrkcond | ex\_swbrkcond | 引数の変更なし | ○ |
| ---------------- | ----------------- | ex\_trgouteventの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_combswの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_m\_corbrkcondの記載はV850E2用EXECからない | × |
| ex\_trcsw | ex\_trcsw | 引数の変更なし | ○ |
| ex\_trcrun | ex\_trcrun | 引数の変更なし | ○ |
| ex\_trcmode | ex\_trcmode | 引数の変更なし  トレース・データ記録条件(tmode2)の設定値固定。  ディレイトリガイベントブレークの削除 | ○ |
| ex\_trcclear | ex\_trcclear | 引数の変更なし | ○ |
| ex\_trcevent | ex\_trcevent | 引数の変更なし  トレース条件(mode)の追加(スレーブ系)および指定方法の変更(設定先種別と条件種別のOR指定) | ○ |
| ex\_trcsearch | ex\_trcsearch | RH850用EXECは未サポート | × |
| ex\_trcseek | ex\_trcseek | RH850用EXECは未サポート | × |
| ex\_trcdelay | ex\_trcdelay | 引数の変更なし  指定するフレーム数の単位変更 | ○ |
| ex\_trcread | ex\_trcread | RH850用EXECは未サポート | × |
| ---------------- | ----------------- | ex\_m\_trcrateの記載はV850E2用EXECからない | × |
| ex\_m\_trceventsel2 | ex\_m\_trceventsel2 | 使用しない(未サポート)  RH850用EXECは未サポート(ex\_rh\_trceventsel2で対応) | × |

表 1‑5 EXEC仕様関連表(3/4)

|  |  |  |  |
| --- | --- | --- | --- |
| V850E2用EXEC関数 | RH850用EXEC関数 | 変更項目 | サポート |
| ---------------- | ex\_rh\_trceventsel2 | RH850用EXECで新設  引数の変更なし  CPUトレース(TCU\_CPU)だけでなく、スレーブトレース(TCU\_TCM, TCU\_LRM, <TCU\_GRM/TCU\_CRM>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, TCU\_GIO, TCU\_EMS)の記録条件(\*item)を追加  Phase1では、MCUの制限によりタイムスタンプ機能をサポートしない。 | ○ |
| ex\_trcbufcont | ex\_trcbufcont | RH850用EXECは未サポート(ex\_rh\_trcbufcontで対応) | × |
| ---------------- | ex\_rh\_trcbufcont | RH850用EXECで新設  RH850用に引数変更し、指定内容も変更 | ○ |
| ex\_trcrawmeminfo | ex\_trcrawmeminfo | 使用方法変更。  totaltimel/totaltimehは未使用  読み出すフレーム数の単位変更  dbg\_mbaパラメータ情報を、ex\_rh\_getenvに移動。  (dbg\_mbaパラメータ自体の定義は残している) | ○ |
| ex\_trcrawmemread | ex\_trcrawmemread | 使用方法変更。  読み出すフレーム数の単位変更 | ○ |
| ex\_trchwtune | ex\_trchwtune | RH850用EXECは未サポート | × |
| ---------------- | ex\_rh\_trchwevncopeinfo | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_trctune | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_trchwtune | RH850用EXECは未サポート  外部トレース時のハードウェア調整機能はex\_rh\_trctuneに実装したため、新設しなかった | × |
| ---------------- | ex\_rh\_sfttrcsw | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrcmode | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrcevent | RH850用EXECで新設 | × |
| ---------------- | ex\_rh\_sfttrcfilter | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrctune | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrcrun | RH850用EXECで新設 | × |
| ---------------- | ex\_rh\_sfttrcrawmemmode | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrcrawmeminfo | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_sfttrcrawmemread | RH850用EXECで新設 | ○ |
| ---------------- | ----------------- | ex\_rrmmappingの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_rrmswの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_rrmreadの記載はV850E2用EXECからない | × |
| ---------------- | ----------------- | ex\_rrmclearの記載はV850E2用EXECからない | × |
| ex\_covmapping | ex\_covmapping | 本開発バージョンではサポートしない  (V3.0以降で検討) | × |
| ex\_covmapping2 | ex\_covmapping2 | × |
| ex\_covsw | ex\_covsw | × |
| ex\_covrun | ex\_covrun | × |
| ex\_covclear | ex\_covclear | × |
| ex\_covsearch | ex\_covsearch | × |
| ex\_covread | ex\_covread | × |
| ex\_covwrite | ex\_covwrite | × |

表 1‑6 EXEC仕様関連表(4/4)

|  |  |  |  |
| --- | --- | --- | --- |
| V850E2用EXEC関数 | RH850用EXEC関数 | 変更項目 | サポート |
| ex\_timesw | ex\_timesw | 引数の変更なし | ○ |
| ex\_timerun | ex\_timerun | 使用しない  RH850では未サポート | × |
| ex\_timeclear | ex\_timeclear | 引数の変更なし | ○ |
| ex\_timeread | ex\_timeread | 使用しない  RH850では未サポート | × |
| ex\_timeread2 | ex\_timeread2 | 引数の変更なし | ○ |
| ex\_timeevent2 | ex\_timeevent2 | 引数の変更なし | ○ |
| ex\_timeevent3 | ex\_timeevent3 | 使用しない  RH850では未サポート | × |
| ex\_timemode | ex\_timemode | 使用しない  RH850では未サポート | × |
| ---------------- | ex\_rh\_timeread | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_timeevent | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_performancesw | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_performanceclear | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_performanceread | RH850用EXECで新設 | ○ |
| ---------------- | ex\_rh\_performanceset | RH850用EXECで新設 | ○ |

## 本仕様書で使用している記述説明と命名規則

### 記述説明

以下に本仕様書で記述する用語について説明する。

|  |  |  |
| --- | --- | --- |
| ICE | | インサーキットエミュレータの略 |
| IE | | インサーキットエミュレータの略 |
| DCU | | デバッグコントロールユニットの略。デバッグ機能マクロの総称 |
|  | RCU | ランコントロールユニットの略。ブレークやバックグラウンドモニタの実行を行うDCUの基本ユニット。RUN-BREAKタイマは~~G3向け~~デバッグ機能のTMUに統合された。 |
| TEU | トリガ・イベント・ユニットの略。多機能イベントを使えるようにするユニット |
| TCU | トレース・コントロール・ユニットの略。トレース機能を使えるようにするユニット |
| MAU | メモリ・アクセス・ユニットの略。 |
| CFU | キャリブレーション機能ユニットの略。 |
| TMU | タイムメジャメントユニットの略。時間計測機能を使える様にするユニット。 |
| RRM | | リアルタイムラムモニタの略。ブレークさせずにメモリの変更を知ることができる。なお、一瞬ブレークさせてデータを読み出す方は、ラムモニタと呼ぶ。 |
| DMM | | 一瞬ブレークさせてデータをモディファイする機能。一部、一瞬ブレークさせてデータを読み出す機能の説明でもDMMと記載しているところもある。 |
| OPBT | | フラッシュメモリ内のExtra領域に設定されるオプションバイト。デバイス関係者用語ではフラッシュマスクオプションと表現する。 |
| BIST | | BiultInSelfTest自己診断機能内蔵 |
| OCD-ID | | OnChipDebug用セキュリティIDコード。~~RH850G3用~~DCUとフラッシュメモリ内のExtra領域を使用したセキュリティ機能のIDコードを示す。 |
| SPF | | SystemProtectionFunction。CPUコア内蔵のシステム保護機能 |
| NEXUS | | IEEE-ISTOで規定されているデバッグ方式。(E1系のN-Wire/N-Trace方式とは異なる) |
| IE850 | | V850E2用 FULL ICE全般を示す。 |
| E2 | | E2 エミュレータを示す。略称はE2と表記する。 |
| IE850A | | IE850Aエミュレータを示す。略称はIE850Aと表記する。 |
| JTAG | | DCUとデバッグ装置のI/Fを示す。NEXUS方式となりRDY端子が追加された。 |
| ERAM(旧TRAM) | | 外部デバッグ装置(開発ツール)からMAU経由でのみアクセス可能な開発ツール専用のメモリ空間であり、コードフラッシュ領域にオーバラップさせて使用することが可能。 |
| BFW | | E1/E20/E2/IE850Aエミュレータ上で動作するファームウェア。本ファームウェアは共通のレベル0部、品種依存のレベルEML部、フラッシュ書き込み用のレベルFDT部の3つから構成される。 |
| モニタ | | RH850~~G3~~コア上で動作するモニタプログラム。 |
| PE | | プロセッサ・エレメントの略。PE1のように番号を付加して特定エレメントを示す。 |
| TEU\_CPU | | RH850~~G3~~コア用のイベント検出器。PEごとに1個搭載する。TEU\_TCM, TEU\_GRM, TEU\_GIOを利用した場合、マスタイベント検出器扱いとなる。 |
| TEU\_TCM | | RH850G3コアのLocalRAM用のイベント検出器。PEごとに1個搭載する。LocalRAM用のスレーブイベント機能を実現するときに用いる。 |
| TEU\_LRM | | RH850G4コアのLocalRAM用のイベント検出器。PEごとに1個搭載する。LocalRAM用のスレーブイベント機能を実現するときに用いる。 |
| TEU\_GRM | | RH850G3コアのGlobalRAM用のイベント検出器。MCUに1個搭載する。GlobalRAM用のスレーブイベント機能を実現するときに用いる。 |
| TEU\_GIO | | RH850G3コアのグローバル周辺IO(VPI)用のイベント検出器。MCUに1個搭載する。VPI用のスレーブイベント機能を実現するときに用いる。 |
| TEU\_CRM | | RH850G4コアのClusterRAM用のイベント検出器。クラスタごとに1個搭載する。ClusterRAM用のスレーブイベント機能を実現するときに用いる。 |
| TEU\_DMA | | RH850G4コアの非CPUマスタ(DMA)用のイベント検出器。MCUに1個搭載する。非CPUマスタ(DMA)のイベント機能を実現するときに用いる。 |

### 命名規則

関数の命名規則を示す。

ただし、EXECデバッグ用内部関数については本命名規則に準ずるものではない。

(1)関数全体の規則

【形式】

ex\_関数名

【説明】

関数名はすべて小文字とする。

(2)関数名の規則

関数名は以下の3種類に分けられる。

(a)単語系(1つの単語で意味が分かるもの)

例)ex\_exit、ex\_run

(b)複合語系(複数の単語を組み合わせて、初めて意味が分かるもの)

単語の組み合わせは、(init･･･ を除き)関数の意味が "aをbする" の場合、ex\_abとする。

例)ex\_memread、ex\_trcsw

その他の関数名の規則を表 1‑7に示す。

表 1‑7 品種別の専用関数名の規則

|  |  |  |  |
| --- | --- | --- | --- |
|  | 関数名 | 意味 | 例 |
|  | init･･･ | IE立ち上げ時に呼ぶ関数 | ex\_initexec |
|  | ･･･cond | イベント条件を設定する関数 | ex\_evncond |
|  | ･･･event | イベント番号を設定する関数 | ex\_trcevent |
|  | ･･･sw | 永久的に切り替える関数 | ex\_trcsw |
|  | ･･･run | 一時的に切り替える関数 | ex\_trcrun |
|  | ･･･clear | データを初期化する関数 | ex\_timeclear |
|  | ･･･read | データを読み出す関数 | ex\_memread |
|  | ･･･write | データを書き込む関数 | ex\_memwrite |

･･･：複合系の単語

(c)品種別専用関数

品種別の専用関数名の規則を表 1‑8に示す。

表 1‑8 品種別の専用関数名の規則

|  |  |  |  |
| --- | --- | --- | --- |
|  | 関数名 | 意味 | 例 |
|  | ex\_m\_･･･ | MIE、BIE専用関数 | ex\_m\_ioread |
|  | ex\_rh\_･･･ | RH850専用関数 | ex\_rh\_evncond |

＜RH850で品種別専用関数を追加する場合の注意＞

・V850E2で既に関数が存在し、RH850で新規に追加する場合は、ex\_rh\_…を使用する。

・RH850で新規に追加する場合は、ex\_…を使用する。

## ヘッダファイル

本仕様書で記載している関数の使用する場合、表 1‑9および表 1‑10に示すヘッダファイルをインクルードする必要がある。

表 1‑9 全品種共通ヘッダファイル一覧

|  |  |
| --- | --- |
| ファイル | 説明 |
| EX\_INC.H | リンク用ヘッダファイル |
| EX\_COM.H | 関数内で使用するマクロ定義ファイル |
| EX\_STRCT.H | 関数で使用する構造体定義ファイル |
| EX\_PROT.H | 関数のプロトタイプ宣言ファイル |
| EX\_ERR.H | 関数より返されるエラーマクロ定義ファイル |

表 1‑10 品種個別ヘッダファイル

|  |  |
| --- | --- |
| ファイル | 説明 |
| EX\_V850.H | 850系(V850E1/ESコア、V850E2コア、RH850 G3コア、RH850 G4コア)のヘッダファイル |

品種個別ヘッダファイルには、各品種専用のマクロ定義、構造体定義、プロトタイプ宣言、エラーマクロ定義が格納されている。

なお、エラーマクロ定義の詳細は、3.3エラーコードを参照のこと。

EXECインターフェース関数利用時は、呼び出しソースに、リンク用ヘッダファイル(EX\_INC.H)のみのインクルードを記述すること。

さらに、ビルドオプションに対象CPUファミリを示すマクロ("EX\_V850")を定義すること。

## インターフェースを提供するプログラムについて

### ソフトウェアの構成および動作環境について(E1/E20)

　E1/E20エミュレータを使用した環境において、EXECインターフェースを提供するプログラムおよびそのプログラムと連動し、デバッグ機能を提供するソフトウェアの構成およびそのソフトウェアが動作する環境を下記に示す。

(1) ホストPC

(2) 統合開発環境(デバッガ)アプリケーション(コンポーネント含む)

(3) EXECインターフェースを提供するプログラム

(7) USB用ドライバ

(11) FDT用のファームウェア

(8) E1/E20エミュレータ

(9) 共通ファームウェア(BFWレベル0)

(4) エミュレータ用情報ファイル

(5) RH850G3系のデバイスファイル

(12) ユーザシステム

(13) RH850G3系マイコン

(14) モニタ

(14) モニタ

(10) RH850G3依存のファームウェア

(6) ルネサス製エミュレータ用USBドライバインターフェース

図 1‑1 ソフトウェアの構成および動作環境について(E1/E20)

「図 1‑1 ソフトウェアの構成および動作環境について(E1/E20)」の説明を下表に示す。

表 1‑11 ソフトウェアの構成および動作環境について

|  |  |
| --- | --- |
| 項番 | 内容 |
| (1) | サポートOS　7(32/64)、8.1(32/64)、10(32/64) |
| (2) | EXECインターフェースを利用するデバッガ機能を含むアプリケーションおよびコンポーネントプログラム |
| (3) | EXECインターフェースを提供するプログラム |
| (4) | E1/E20エミュレータの情報を定義したファイル |
| (5) | エミュレーション対象となるRH850G3系マイコン用情報のファイル(サポート品種ごとに個別に存在する) |
| (6) | ルネサス製エミュレータ用USBドライバインターフェース(Communi.dll) |
| (7) | E1/E20エミュレータ用のUSBドライバプログラム(SYS) |
| (8) | E1/E20エミュレータハードウェア |
| (9) | E1/E20エミュレータ用共通のファームウェア(通信部基本機能) |
| (10) | E1/E20エミュレータ用RH850G3依存のファームウェア(BFWレベルEML) |
| (11) | E1/E20エミュレータ用RH850G3依存のFDTファームウェア |
| (12) | RH850G3系マイコンを搭載したユーザシステム |
| (13) | エミュレーション対象のRH850G3系マイコン |
| (14) | RH850G3系マイコン上で動作するモニタプログラム |

### ソフトウェアの構成および動作環境について(E2/IE850A)

　E2エミュレータ/IE850Aエミュレータを使用した環境において、EXECインターフェースを提供するプログラムおよびそのプログラムと連動し、デバッグ機能を提供するソフトウェアの構成およびそのソフトウェアが動作する環境を下記に示す。

(1) ホストPC

(2) 統合開発環境(デバッガ)アプリケーション(コンポーネント含む）

(3) EXECインターフェースを提供するプログラム

(7) USB用ドライバ

(11) RFP用のファームウェア

(8) E2エミュレータ/IE850Aエミュレータ

(9) 共通ファームウェア(BFWレベル0)

(4) エミュレータ用情報ファイル

(5) RH850G3/G4系のデバイスファイル

(12) ユーザシステム

(13) RH850G3/G4系マイコン

(14) モニタ

(14) モニタ

(10) RH850G3/G4依存のファームウェア

(6) USBドライバインターフェースプログラム

図 1‑2 ソフトウェアの構成および動作環境について(E2/IE850A)

「図 1‑2 ソフトウェアの構成および動作環境について(E2/IE850A)」の説明を下表に示す。

表 1‑12 ソフトウェアの構成および動作環境について

|  |  |
| --- | --- |
| 項番 | 内容 |
| (1) | サポートOS　7(32/64)、8.1(32/64)、10(32/64) |
| (2) | EXECインターフェースを利用するデバッガ機能を含むアプリケーションおよびコンポーネントプログラム |
| (3) | EXECインターフェースを提供するプログラム |
| (4) | E2エミュレータ/IE850Aエミュレータの情報を定義したファイル |
| (5) | エミュレーション対象となるRH850G3/G4系マイコン用情報のファイル(サポート品種ごとに個別に存在する) |
| (6) | ルネサス製エミュレータ用USBドライバインターフェース(Communi.dll) |
| (7) | E2エミュレータ/IE850Aエミュレータ用のUSBドライバプログラム(SYS) |
| (8) | E2エミュレータ/IE850Aエミュレータハードウェア |
| (9) | E2エミュレータ/IE850Aエミュレータ用共通のファームウェア(通信部基本機能) |
| (10) | E2エミュレータ/IE850Aエミュレータ用RH850G3/G4依存のファームウェア(BFWレベルEML) |
| (11) | E2エミュレータ/IE850Aエミュレータ用RH850G3/G4依存のRFPファームウェア |
| (12) | RH850G3/G4系マイコンを搭載したユーザシステム |
| (13) | エミュレーション対象のRH850G3/G4系マイコン |
| (14) | RH850G3/G4系マイコン上で動作するモニタプログラム |

### ソフトウェアの構成および動作環境について(IE850)

　IE850エミュレータを使用した環境において、EXECインターフェースを提供するプログラムおよびそのプログラムと連動し、デバッグ機能を提供するソフトウェアの構成およびそのソフトウェアが動作する環境を下記に示す。

(1) ホストPC

(2) 統合開発環境(デバッガ)アプリケーション(コンポーネント含む)

(3) EXECインターフェースを提供するプログラム

(6) USB用ドライバ

(7) IE850エミュレータ

(8) ファームウェア(CTファーム)

(4) エミュレータ用情報ファイル

(5) RH850G3系のデバイスファイル

(9) ユーザシステム

(10) RH850G3系マイコン

(11) モニタ

(11) モニタ

(12) 本体VERアップツール

図 1‑3 ソフトウェアの構成および動作環境について(IE850)

「図 1‑3 ソフトウェアの構成および動作環境について(IE850)」の説明を下表に示す。

表 1‑13 ソフトウェアの構成および動作環境について

|  |  |
| --- | --- |
| 項番 | 内容 |
| (1) | サポートOS　7(32/64)、8.1(32/64)、10(32/64) |
| (2) | EXECインターフェースを利用するデバッガ機能を含むアプリケーションおよびコンポーネントプログラム |
| (3) | EXECインターフェースを提供するプログラム |
| (4) | IE850エミュレータの情報を定義したファイル |
| (5) | エミュレーション対象となるRH850G3系マイコン用情報のファイル(サポート品種ごとに個別に存在する) |
| (6) | IE850エミュレータ用のUSBドライバプログラム(SYS) |
| (7) | IE850エミュレータハードウェア |
| (8) | RH850G3向けIE850エミュレータ用ファームウェア(CTファーム) |
| (9) | RH850G3系マイコンを搭載したユーザシステム |
| (10) | エミュレーション対象のRH850G3系マイコン |
| (11) | RH850G3系マイコン上で動作するモニタプログラム |
| (12) | IE850本体のFPGAデータ/ファームウェアのVERアップを行う専用ツール |

### 動作環境についての注意点

動作環境として使用できるエミュレータはE1エミュレータ、E20エミュレータ、E2エミュレータ、IE850エミュレータ、IE850Aエミュレータの5種類のみとする。また、1台のホストPCに対して同時に接続可能なエミュレータはE1/E20エミュレータで合計4台まで、E2/E2 Lite/IE850Aエミュレータは4台まで、IE850は4台までとする(E1/E20エミュレータとE2/E2 Lite/IE850AエミュレータとIE850エミュレータの同時接続台数の合計が4台を超えた場合でも、E1/E20エミュレータの合計接続台数が4台以下、E2/E2 Lite/IE850Aエミュレータの接続台数が4台以下、IE850エミュレータの接続台数が4台以下の場合は接続可能)。

E1/E20/E2/E2 Liteエミュレータは、RXやRLと共通であるが、RXのFFWやRLのEXECで接続されているエミュレータも1台としてカウントされる。他のFFWやEXEC、ユーティリティと接続されているかどうかに関わらず、PCに接続されているエミュレータすべてがカウント対象となる。

RH850ではE2 Liteはサポートしていないが、E2/E2 Lite/IE850Aは同じUSBドライバであるため同一としてカウントされる。そのため先にE2/E2 Lite/IE850A 合計で4台以上PCに接続されている場合は、接続不可となるので注意すること。

RH850G3世代EXECのV4.04.00.00以降、および、RH850G4世代EXECのV1.02.00.00以降で64bitアプリの対応を行う。既存の32bitアプリと区別するため、モジュール名に「\_x64」を付加する。EXECの64bitアプリ対応に伴い、Communiも64bitアプリ対応を行う。その他のモジュールは影響を受けないため、既存の32bit版/64bit版に関わらず共通で使用する。EXECの階層においては、32bit版と64bit版で機能差はない。

### プログラムの構成について

EXECインターフェースを提供するプログラムはWindowsのダイナミックライブラリ(DLL)形式とする。

本プログラムの構成を下表に示す。

表 1‑14 プログラムの構成

|  |  |
| --- | --- |
| 項目 | 説明 |
| ファイル名称 | EXRH850G3.DLL (RH850G3世代内で共通)  EXRH850G3\_x64.DLL (RH850G3世代内で共通)  EXRH850G4.DLL (RH850G4世代内で共通)  EXRH850G4\_x64.DLL (RH850G4世代内で共通) |
| .NET Frameworkの依存 | 依存しない |
| MFCについて | MFCは使用しないで、標準Windowsライブラリを使用する。 |

### EXRH850Gx.DLLファイルのプロパティについて

ファイルのプロパティの表示内容を下表に示す。

表 1‑15 EXRH850G3.DLLファイルのプロパティ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目 | 表示例(32bit版) | 表示例(64bit版) | 説明 |
| Comments | V4.04.00.xx [dd mmm yyyy] | | xx：リリース単位に01->99に順に変更する  dd：日付  mmm：月  yyyy：年 |
| CompanyName | Renesas Electronics Corporation | | 左記固定 |
| FileDescription | RH850G3 Executor DLL | [64bit] RH850G3 Executor DLL | 左記固定 |
| FileVersion | 4.04.00.xx | | xx：リリース単位に01->99に順に変更する |
| InternalName | EXRH850G3.dll | EXRH850G3\_x64.dll | 左記固定 |
| LegalCopyright | © 2012,yyyy Renesas Electronics Corporation | | yyyy：年 |
| OriginalFilename | EXRH850G3.dll | EXRH850G3\_x64.dll | 左記固定 |
| ProductName | EXRH850G3 | | 左記固定 |
| ProductVersion | 4.04.00.00 | | 左記固定 |

表 1‑16 EXRH850G4.DLLファイルのプロパティ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目 | 表示例(32bit版) | 表示例(64bit版) | 説明 |
| Comments | 1.02.00.xx [dd mmm yyyy] | | xx：リリース単位に01->99に順に変更する  dd：日付  mmm：月  yyyy：年 |
| CompanyName | Renesas Electronics Corporation | | 左記固定 |
| FileDescription | RH850G4 Executor DLL | [64bit] RH850G4 Executor DLL | 左記固定 |
| FileVersion | 1.02.00.xx | | xx：リリース単位に01->99に順に変更する |
| InternalName | EXRH850G4.dll | EXRH850G4\_x64.dll | 左記固定 |
| LegalCopyright | © 2017,yyyy Renesas Electronics Corporation | | yyyy：年 |
| OriginalFilename | EXRH850G4.dll | EXRH850G4\_x64.dll | 左記固定 |
| ProductName | EXRH850G4 | | 左記固定 |
| ProductVersion | 1.02.00.00 | | 左記固定 |

### 提供ファイルについて

　EXECインターフェースを提供するプログラムを動作させるために必要なファイルセットの一覧を下表に示す。

表 1‑17 ファイルセット一覧

|  |  |
| --- | --- |
| ファイル名 | 説明 |
| EXRH850G3.DLL | RH850G3用EXECの本体(プログラム) (32bit版) |
| EXRH850G3\_x64.DLL | RH850G3用EXECの本体(プログラム) (64bit版) |
| EXRH850G3.INI | RH850G3用INI定義ファイル |
| EXRH850G4.DLL | RH850G4用EXECの本体(プログラム) (32bit版) |
| EXRH850G4\_x64.DLL | RH850G4用EXECの本体(プログラム) (64bit版) |
| EXRH850G4.INI | RH850G4用INI定義ファイル |
| RH850Emulator.env | エミュレータ部品の整合性チェックを行うための管理情報定義ファイル |
| BfwE20RH850G3.s | RH850G3 E1/E20用BFWプログラム(レベルEML) |
| BfwE2RH850.s | RH850G3/G4 E2用BFWプログラム(レベルEML) |
| BfwIE850ARH850.s | RH850G4 IE850A用BFWプログラム(レベルEML) |
| e1\_v850lpd.sbt | RH850G3 E1用のFPGAデータ |
| e20\_v850lpd.sbt | RH850G3 E20用のFPGAデータ |
| E2\_RH850.bit | RH850G3/G4 E2用のFPGAデータ |
| IE850A\_1L\_2B.bit | RH850G4 IE850A用のFPGAデータ(Aurora1レーン/2バイト用) |
| IE850A\_4L\_2B.bit | RH850G4 IE850A用のFPGAデータ(Aurora4レーン/2バイト用) |
| monpprog.s | E1/E20エミュレータ上にあるH8Sマイコンの内蔵フラッシュメモリに書き込まれたBFWプログラムを書きかえるためのプログラム。 |
| MONPPROG\_E2.s | E2エミュレータ上にあるRXマイコンの内蔵フラッシュメモリに書き込まれたBFWプログラムを書き換えるためのプログラム。 |
| Communi.dll | E1/E20/E2/IE850AエミュレータのUSB通信インターフェース(32bit版) |
| Communi\_x64.dll | E1/E20/E2/IE850AエミュレータのUSB通信インターフェース(64bit版) |

# インターフェース仕様

## インターフェース関数

### インターフェース関数の分類

[初期化関数]

起動時における初期化、終了、デバイスファイルロード、リセットのための関数である。

ex\_exit ex\_initexec ex\_reset ex\_setemulator ex\_getemulist ex\_extpower

[ハードウェア条件設定-状態検出関数]

IEの状態を監視するための関数である。

ex\_getenv ex\_rh\_getenv ex\_getstatus ex\_setcallback\_sts ex\_boardinfo ex\_boardinfo2

[ハードウェア条件設定-環境設定関数]

環境設定をするための関数である。

ex\_setenv ex\_initimem ex\_clock ex\_mapping

ex\_pincont ex\_m\_montimeout

ex\_socunitinfo ex\_socunitinfo2

[メモリ制御関数]

メモリ(SFR、拡張SFR空間も含む)を操作するための関数である。

ex\_memread ex\_memwrite ex\_memfill ex\_memcopy ex\_memsearch ex\_memcompare

[レジスタ制御関数]

レジスタを操作するための関数である。

ex\_regread ex\_regwrite

[イベント制御関数]

イベント設定参照のための関数である。

ex\_evncond ex\_evnintcond

ex\_rh\_evncond ex\_rh\_evnintgcond

[実行制御関数]

ユーザプログラム実行を制御するための関数である。

ex\_run ex\_resetrun ex\_step ex\_stop ex\_hwbrkevent ex\_swbrkcond

[トレース制御関数]

トレースを制御するための関数である。

[Trace control function]  
   It is a function for controlling the trace.

ex\_trcsw ex\_trcrun ex\_trcmode ex\_trcclear ex\_trcevent ex\_trcsearch ex\_trcseek

ex\_trcdelay ex\_trcread ex\_m\_trceventsel2 ex\_rh\_trceventsel2 ex\_ m\_trcbufcont

ex\_rh\_trcbufcont ex\_trcrawmeminfo ex\_trcrawmemread ex\_trchwtune

ex\_rh\_trchwevncopeinfo ex\_rh\_trctune

[外部ソフトトレース単線出力制御関数]

外部ソフトトレース単線出力を制御するための関数である。

[External soft trace single line output control function]  
This is a function to control the external soft trace single line output.

ex\_rh\_sfttrcsw ex\_rh\_sfttrcmode ex\_rh\_sfttrcevent ex\_rh\_sfttrcfilter　ex\_rh\_sfttrctune

ex\_rh\_sfttrcrun ex\_rh\_sfttrcrawmemmode ex\_rh\_sfttrcrawmeminfo ex\_rh\_sfttrcrawmemread

[RRM制御関数]

リアルタイムラムモニタを制御するための関数である。

ex\_rrmmapping ex\_rrmsw ex\_rrmread ex\_rrmclear

[カバレッジ制御関数]

カバレッジを制御するための関数である。

ex\_covmapping ex\_covmapping2 ex\_covsw ex\_covrun ex\_covclear ex\_covread ex\_covwrite

ex\_covsearch

[タイマ関数/パフォーマンス計測関数]

タイマとパフォーマンス計測を制御するための関数である。

ex\_timesw ex\_timerun ex\_timeclear ex\_timeread ex\_timeread2 ex\_timeevent2 ex\_timeevent3

ex\_timemode ex\_rh\_timeread ex\_rh\_timeevent ex\_rh\_performancesw ex\_rh\_performanceclear

ex\_rh\_performanceread ex\_rh\_performanceset

[非同期通信処理関数]

非同期通信処理を行う専用の関数である。

callback\_sts

### インターフェース関数一覧

表 2‑1～表 2‑4にインターフェース関数の一覧表を示す。

×が付いている関数は、RH850ではサポートしていないため、EX\_NOSUPRTを返す。

ただし、接続方式によっては、接続が確立するまではEX\_FE\_COMMが返る。

表にない場合は、EX\_NOSUPRTを返す関数エントリ自体がないことを表す。

EXEC I/Fでは、未サポートのパラメータが存在する。未サポートのパラメータは、0クリアを行ってからEXEC I/Fを発行すること。

表 2‑1 インターフェース関数一覧(1/4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I/F名 | | 説明 | サポート  状況 | |
| G3 | G4 |
| ex\_exit | | EXECの終了を行う | ○ | ○ |
| ex\_initexec | | IEとの通信をオープンし、各環境の初期化とEXECの起動を行う | ○ | ○ |
| ex\_reset | | エミュレーションCPUのリセットを行う | ○ | ○ |
| ex\_setemulator | | 接続するICEの種類やシリアル文字列を指定する | ○ | ○ |
| ex\_getemulist | | PCに接続されているエミュレータのリストを取得する | ○ | ○ |
| ex\_extpower | | E2エミュレータ/IE850Aエミュレータの電源供給設定を行う | ○ | ○ |
| ex\_getenv | | 各種環境を参照する | ○ | ○ |
| ex\_rh\_getenv | | デバッグ機能のサポート状況を参照する | ○ | ○ |
| ex\_getstatus | | 各種ステータス情報を読み出す | ○ | ○ |
| ex\_setcallback\_sts | | 非同期通信用コールバック関数のアドレス情報の設定/参照 | ○ | ○ |
| ex\_boardinfo | | ボードバージョンを得る | × | × |
| ex\_boardinfo2 | | ボードバージョンを得る | ○ | ○ |
| ex\_setenv | | IEの各種環境を設定/参照する | ○ | ○ |
| ex\_initimem | | 内部ROMのサイズと内部RAMのサイズ、ロケーションを設定する | × | × |
| ex\_clock | | クロックを設定/参照する | × | × |
| ex\_mapping | | 外部メモリのマッピング状態を設定する | × | × |
| ex\_pincont | | 端子の設定/参照を行う | ○ | ○ |
| ex\_m\_ieport | | IE用汎用ポートの設定/参照を行う | ○ | ○ |
| ex\_m\_montimeout | | モニタタイムアウトの時間を設定する | ○ | ○ |
| ex\_socunitinfo | | SOCデバイス内蔵ユニットの情報を設定/参照する | × | × |
| ex\_socunitinfo2 | | SOCデバイス内蔵ユニットの情報を設定/参照する(拡張版) | ○ | ○ |
|  | EX\_UNIT\_LPDOPT | LPD接続の設定 | ○ | ○ |
|  | EX\_UNIT\_RSU | OCDセキュリティIDコードの解除/参照 | ○ | ○ |
|  | EX\_UNIT\_CG | クロック関係の設定 | ○ | ○ |
|  | EX\_UNIT\_FLNUMINFO | コードフラッシュブロック個数情報取り出し | × | × |
|  | EX\_UNIT\_FLFUNCINFO | コードフラッシュ機能情報参照 | ○ | ○ |
|  | EX\_UNIT\_FLMAPINFO | コードフラッシュマッピング情報参照 | ○ | ○ |
|  | EX\_UNIT\_FLBLOCKINFO | コードフラッシュブロックアドレス情報取り出し | ○ | ○ |
|  | EX\_UNIT\_DFLNUMINFO | データフラッシュブロック個数情報取り出し | ○ | ○ |
|  | EX\_UNIT\_DFLMAPINFO | データフラッシュマッピング情報参照 | ○ | ○ |
|  | EX\_UNIT\_DFLBLOCKINFO | データフラッシュブロックアドレス情報取り出し | ○ | ○ |
|  | EX\_UNIT\_DFLAREABLOCKINFO | 領域別データフラッシュブロックアドレス情報取り出し | ○ | ○ |
|  | EX\_UNIT\_DCUSEC | OCDセキュリティIDコードの設定 | × | × |
|  | EX\_UNIT\_FLMKOP | フラッシュマスクオプション設定 | ○ | × |
|  | EX\_UNIT\_FLSECINFO | コードフラッシュのExtra領域(SecurityFlag)の設定/参照 | × | × |

表 2‑2 インターフェース関数一覧(2/4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I/F名 | | 説明 | サポート  状況 | |
| G3 | G4 |
|  | EX\_UNIT\_FLEXTRAINIT | コードフラッシュのExtra(フラッシュマスクオプション、SecurityFlag、LockBit)領域の初期化 | × | × |
|  | EX\_UNIT\_FLLBINFO | コードフラッシュのExtra領域(LockBit)の設定/参照 | 〇 | × |
|  | EX\_UNIT\_DFLBLNKINFO | データフラッシュブランク状態参照 | 〇 | 〇 |
|  | EX\_UNIT\_FID | フラッシュメモリ用セキュリティIDコードの認証指定/  解除状態の参照 | 〇 | × |
|  | EX\_UNIT\_COREPESEL | 操作対象PE(NM)/VM/HT番号の設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_SWBRKFUNCTION | フラッシュメモリへのソフトウェアブレーク設定方式の設定 | 〇 | 〇 |
|  | EX\_UNIT\_EXTFLINFO | 未使用(外付けフラッシュ情報) | × | × |
|  | EX\_UNIT\_RUNCONTOPT | ex\_run()の状態継続の設定 | 〇 | 〇 |
|  | EX\_UNIT\_DMMRRMOPT | DMM、RRMの動作設定 | 〇 | 〇 |
|  | EX\_UNIT\_FLPGOPT | フラッシュメモリへの書き込み保護動作関連の設定 | 〇 | 〇 |
|  | EX\_UNIT\_TRCSEL | 使用するトレース出力タイプの設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_SLAVEIDSEL | 操作対象スレーブIDの設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_VIRTUALINFO | 仮想化支援機能使用有無の設定/参照 | 〇 | × |
|  | EX\_UNIT\_FLCCROPT | フラッシュメモリ書き換え時のクロック制御設定 | 〇 | 〇 |
|  | EX\_UNIT\_ASYNCPEOPT | 複数PE間の非同期デバッグモードの設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_TRACEOPT | トレース関連のオプション設定 | 〇 | 〇 |
|  | EX\_UNIT\_TRACEMSGPEID | トレースメッセージに記録されるPEIDの値を設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_WORKRAM | ワークRAM領域の開始アドレスを設定/参照  ワークRAM領域のサイズを参照 | 〇 | 〇 |
|  | EX\_UNIT\_DBGINFOUT | 任意のデバッグ情報の設定/参照 | 〇 | 〇 |
|  | EX\_UNIT\_CMFID | 起動時に必ず認証が必要IDコード(Customer ID/Data Flash ID)の認証/認証解除状態の参照 | × | 〇 |
|  | EX\_UNIT\_OPTID | 任意で認証するIDコード(Customer ID/Data Flash ID以外)の認証/認証解除状態の参照 | × | 〇 |
|  | EX\_UNIT\_CLUSTERSEL | 操作対象クラスタ番号の設定/参照 | × | 〇 |
|  | EX\_UNIT\_EXTTRACE | 外部トレースの転送速度の設定/参照 | × | 〇 |
|  | EX\_UNIT\_SVR | SVRパラメータ設定 | × | ○ |
|  | EX\_UNIT\_EFRECOVERY | SWASリカバーコマンド実行 | × | ○ |
|  | EX\_UNIT\_HEGESEL | 仮想化支援機能使用時のデバッグ対象の設定/参照 | × | ○ |
|  | EX\_UNIT\_INITSTOPDBG | 初期停止・スタンバイモードデバッグ環境の設定/参照 | ○ | ○ |
| ex\_memread | | メモリからデータを読み出す | ○ | ○ |
| ex\_memwrite | | メモリへデータを書き込む | ○ | ○ |
| ex\_memfill | | メモリを初期化する | ○ | ○ |
| ex\_memcopy | | メモリ・データをコピーする | ○ | ○ |
| ex\_memsearch | | メモリのデータを検索する | × | × |
| ex\_memcompare | | メモリ・データを比較する | × | × |
| ex\_regread | | レジスタのデータを読み出す | ○ | ○ |
| ex\_regwrite | | レジスタへデータを書き込む | ○ | ○ |
| ex\_sfrread | | SFRのデータを読み出す | × | × |
| ex\_sfrwrite | | SFRへデータを書き込む | × | × |
| ex\_evncond | | イベント条件を設定/参照/削除する | × | × |
| ex\_evnintgcond | | イベント統合条件を設定/参照/削除する | × | × |
| ex\_rh\_evncond | | イベント条件を設定/参照/削除する | ○ | ○ |
| ex\_rh\_evnintgcond | | イベント統合条件を設定/参照/削除する | ○ | ○ |
| ex\_run | | ユーザプログラムを実行する | ○ | ○ |
| ex\_resetrun | | リセット付きでユーザプログラムを実行する | ○ | ○ |
| ex\_step | | ユーザプログラムをステップ実行する | ○ | ○ |
| ex\_stop | | ユーザプログラムを強制ブレークする | ○ | ○ |
| ex\_hwbrkevent | | イベントブレークを設定/参照/削除する | ○ | ○ |
| ex\_swbrkcond | | ソフトウェアブレークを設定/参照/削除する | ○ | ○ |
| ex\_trgoutevent | | トリガアウト・イベントを設定/参照/削除する | × | × |

表 2‑3 インターフェース関数一覧(3/4)

|  |  |  |  |
| --- | --- | --- | --- |
| I/F名 | 説明 | サポート  状況 | |
| G3 | G4 |
| ex\_combsw | コンボブレーク機能を有効/無効にする | × | × |
| ex\_m\_corbrkcond | コアブレークイベント条件の詳細設定を行う | × | × |
| ex\_trcsw | トレース機能を有効/無効にする | ○ | ○ |
| ex\_trcrun | プログラム実行中のトレース開始/終了を制御する | ○ | ○ |
| ex\_trcmode | トレースモードを設定/参照する | ○ | ○ |
| ex\_trcclear | トレース・カウンタをクリアする | ○ | ○ |
| ex\_trcevent | トレースの各種イベント条件を設定/参照/削除する | ○ | ○ |
| ex\_trcsearch | トレース・データの検索条件の設定/参照を行う | × | × |
| ex\_trcseek | トレース・ポインタを移動する | × | × |
| ex\_trcdelay | ディレイイベント発生時のディレイ値を設定/参照する | ○ | ○ |
| ex\_trcread | トレース・データを読み出す | ○ | ○ |
| ex\_m\_trcrate | トレース・タイムタグ・カウンタの分周率を設定/参照する | × | × |
| ex\_m\_trceventsel2 | トレースフレームの有効データを設定/参照する | × | × |
| ex\_rh\_trceventsel2 | トレースフレームの有効データを設定/参照する | ○ | ○ |
| ex\_m\_trcbufcont | TCUのトレースバッファを設定/参照する | × | × |
| ex\_rh\_trcbufcont | TCUのトレースバッファを設定/参照する | ○ | ○ |
| ex\_trcrawmeminfo | トレース・生データリードに必要な情報を取得する | ○ | ○ |
| ex\_trcrawmemread | トレース・生データを読み出す | ○ | ○ |
| ex\_trchwtune | トレース・データ取得関連のハードウェア設定/参照する | × | × |
| ex\_rh\_trchwevncopeinfo | トレース起動時の単体イベント番号に対応するハードウェアイベント番号情報の参照 | ○ | ○ |
| ex\_rh\_trctune | トレースモジュール(TCU)の調整機能 | ○ | ○ |
| ex\_rh\_trchwtune | 外部トレース時のハードウェア調整機能 | × | × |
| ex\_rh\_trcfilter | トレースのフィルタ条件を設定/参照する | ×注１ | ○ |
| ex\_rh\_sfttrcsw | 外部ソフトトレース単線出力機能を有効/無効に設定/参照する  Enable / disable external soft trace single-wire output function | ○ | ○ |
| ex\_rh\_sfttrcmode | 外部ソフトトレース単線出力機能の動作状態を設定/参照する  Sets / references the operating status of the external soft trace single-wire output function | ○ | ○ |
| ex\_rh\_sfttrcevent | 外部ソフトトレース単線出力機能のイベント条件を設定/参照する  Set / reference the event condition of the external soft trace single line output function | × | × |
| ex\_rh\_sfttrcfilter | 外部ソフトトレース単線出力機能のフィルタ条件を設定/参照する  Set / reference the filter condition of the external soft trace single line output function | ○ | ○ |
| ex\_rh\_sfttrctune | ソフトトレースモジュール(TCU\_SFT)の調整機能  Adjustment function of soft trace module (TCU\_SFT) | ○ | ○ |
| ex\_rh\_sfttrcrun | 外部ソフトトレース単線出力の記録動作を設定/参照する。  Sets / references the recording operation of external soft trace single line output. | × | × |
| ex\_rh\_sfttrcrawmemmode | 外部ソフトトレース単線出力生データ用メモリの動作状態を設定／参照する。  Sets / references the operating status of the external soft trace single-wire output raw data memory. | ○ | ○ |
| ex\_rh\_sfttrcrawmeminfo | 外部ソフトトレース単線出力生データリードに必要な情報を得る。  Obtain information necessary for reading raw data of external soft trace single wire output. | ○ | ○ |
| ex\_rh\_sfttrcrawmemread | 外部ソフトトレース単線出力生データリード  External soft trace single wire output raw data read | ○ | ○ |
| ex\_rrmmapping | RRMの領域設定を行う | × | × |
| ex\_rrmsw | RRMの有効無効の設定を行う | × | × |
| ex\_rrmread | RRMの情報を読み出す | × | × |
| ex\_rrmclear | RRMの情報をクリアする | × | × |
| ex\_covmapping2 | カバレッジ・メモリの領域設定/参照を行う | × | × |
| ex\_covmapping | カバレッジ・メモリの領域設定/参照を行う | × | × |
| ex\_covsw | カバレッジ・メモリを有効/無効にする | × | × |
| ex\_covrun | カバレッジ測定の強制開始/終了を行う | × | × |
| ex\_covclear | カバレッジ測定結果を初期化(クリア)する | × | × |
| ex\_covsearch | 指定したアドレス範囲のカバレッジデータを検索する | × | × |
| ex\_covread | カバレッジ測定結果を読み出す | × | × |
| ex\_covwrite | カバレッジ測定結果をカバレッジ・メモリに書き込む | × | × |

注1：関数のエントリ自体がない

表 2‑4 インターフェース関数一覧(4/4)

|  |  |  |  |
| --- | --- | --- | --- |
| I/F名 | 説明 | サポート  状況 | |
| G3 | G4 |
| ex\_timesw | タイマの有効/無効を切り替える | ○ | ○ |
| ex\_timerun | タイマの強制開始/終了を行う | × | × |
| ex\_timeclear | タイマの計測結果を初期化する | ○ | ○ |
| ex\_timeread | タイマの計測結果を読み出す | × | × |
| ex\_timemode | タイマのモードを設定/参照する | × | × |
| ex\_timeread2 | タイマの計測結果を読み出す | ○ | ○ |
| ex\_timeevent2 | TMU\_CLKタイマの計測条件を設定/参照する | ○ | ○ |
| ex\_timeevent3 | タイマのイベント条件を設定/参照する | × | × |
| ex\_rh\_timeread | TMU\_TCKタイマの計測結果を読み出す | ○ | ○ |
| ex\_rh\_timeevent | TMU\_TCKタイマの計測条件を設定/参照する | ○ | ○ |
| ex\_rh\_performancesw | パフォーマンス計測の有効/無効を切り替える | ○ | ○ |
| ex\_rh\_performanceclear | パフォーマンス計測結果を初期化する | ○ | ○ |
| ex\_rh\_performanceread | パフォーマンス計測結果を読み出す | ○ | ○ |
| ex\_rh\_performanceset | パフォーマンス計測条件を設定/参照する | ○ | ○ |

## 起動と終了

### 起動

起動は、ex\_boardinfo2でエミュレータのコンフィグレーションを行う。この時、一部デバイス情報が必要となる場合があるので、デバイスファイルやエミュレータ-デバイス間の接続に関する指定が必要となる。

続いて、ex\_initexec関数でデバイス情報の読み込みや各ICEのハードウェアの初期化を行う。

ICEとの通信が行えない場合は、エラー(EX\_FE\_COMM)を返す。

エミュレータの指定を行う前に、どんなエミュレータがPCに接続されているか確認する場合は、最初にex\_getemulist()を呼び出すこと(確認の必要がない場合は以下の(1)は省略可能)。

接続するエミュレータの指定を行う際は必ず(3)の前にex\_setemulator()を呼び出すこと(エミュレータの指定を行わない場合は以下の(2)は省略可能)。

E2エミュレータ/IE850Aエミュレータ専用の電源供給設定を行う場合は(6)の前にex\_extpower()を呼び出すこと(設定が不要な場合は以下の(5)は省略可能)。

G4世代MCUは、ユーザプログラムが自走しないで起動できる機能を持っている。この場合、ユーザプログラムが自走しない手順固定で起動処理を行なう。

G4世代MCUの場合、エミュレータ未接続時は初期値が不定である以下のレジスタを、エミュレータ接続時は起動処理完了時に初期化する(不定状態でのレジスタ読み出しによるECMロックステップエラーを抑止するため)。なお、レジスタの初期化は通常起動時でかつロックステップ搭載コアのみ行ない、ロックステップ未搭載コアおよびホットプラグイン接続時は行わない。

初期化するレジスタ：R1、R3、R[31:6]、EIPC、デバッグ用レジスタ

初期化する値　　　：0x00000000

[デバッガ関係者向け注意事項]

起動において通信エラー(EX\_FE\_COMM)、USB通信エラー(EX\_FE\_USBFAIL)、冷却ファンエラー(EX\_FE\_FANERROR)、電源供給エラー(EX\_FE\_VOUTERROR,EX\_FE\_BEFOREVOUTERROR)またはターゲット電源OFFエラー(EX\_SE\_TRGTOFF)が発生し、以降のI/Fでも上記エラーのいずれかが出続ける状態になった場合、デバッガは起動を中止すること。

＜G3世代MCU起動時に必要なEXEC関数の手順(通常起動)＞

(1) ex\_getemulist()★

(2) ex\_setemulator()★

(3) ex\_boardinfo2()

＜ターゲット接続時の注意事項＞

ターゲット電源OFF時は、電源ON後に再度ex\_boardinfo2()を呼び出す必要がある

(4) ex\_setcallback\_sts()★

＜注意事項＞

呼び出す必要があるのは、非同期通信I/Fを使用する場合のみ。

(5) ex\_extpower()★

＜注意事項＞

E2エミュレータ以外では呼び出し禁止。E2エミュレータでは呼び出さなくても起動は可能だが、E2エミュレータサポート時は本I/Fを呼び出すようにすること。

(6) ex\_initexec()

(7) ex\_socunitinfo2() unit\_type=EX\_UNIT\_CG

(8) ex\_socunitinfo2() unit\_type=EX\_UNIT\_INITSTOPDBG★

※デバイスファイルのEFIタグでfunc:0x0103＜エミュレーション機能情報3＞の

bit3(Initial Stop Debug)が1の場合、初期停止・スタンバイモードデバッグ環境の有効/無効を選択することが可能である。呼び出しをしなかった場合は、無効が選択される。

詳細は、ex\_socunitinfo2の[Description]を参照する事。

(9) ex\_socunitinfo2() unit\_type=EX\_UNIT\_LPDOPT

※デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合、端子リセットディレイ対策の選択を行う必要がある。詳細は、ex\_socunitinfo2の[Description]を参照する事。

(10) ex\_socunitinfo2() unit\_type= EX\_UNIT\_FID

※DFファイル内EFIタグ「エミュレーション機能情報1」のbit18(CodeFlash-ID)、bit19(DataFlash-ID)の各情報が‘1’の場合は、本コマンドの発行が必要である。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、ex\_socunit\_info2()unit\_type=EX\_UNIT\_RSUのunit\_info[4](unit\_info[8])、あるいは起動手順完了後(注1)に本コマンドを参照指定で発行することで確認できる。

(11) ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU

　必要であれば、(8)～(11)を繰り返す

(12) ex\_socunitinfo2() unit\_type=EX\_UNIT\_COREPESEL

デバイスファイルで有効なPE数分だけ(12)を繰り返し、無効なPEがないか確認する。

(13) ex\_pincont()

(14) ex\_getenv()

(15) ex\_rh\_getenv()

★のI/Fの呼び出しは必須ではない。呼び出さなくても起動する。

注1：起動手順完了後とは、ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU発行後のことである。

(ICUM有効モードでかつC&R認証が必要な場合を除く)

＜G3世代MCUホットプラグイン起動の手順(OCDのみ)＞

(1) ex\_getemulist()★

(2) ex\_setemulator()★

(3) ex\_boardinfo2() ： ホットプラグインアダプタにE1/E20/E2ユーザインターフェースケーブルが未接続

(ステータス情報からターゲット電源OFF)を確認する

(4) ex\_setcallback\_sts()★

※呼び出す必要があるのは、非同期通信I/Fを使用する場合のみ。

(5) ex\_extpower()★ ： ホットプラグイン起動指定で呼び出す

＜注意事項＞

E2エミュレータ以外では呼び出し禁止。E2エミュレータでは呼び出さなくても起動は可能だが、E2エミュレータサポート時は本I/Fを呼び出すようにすること。

(6) ex\_initexec() ： ホットプラグイン起動指定で呼び出す

ユーザにE1/E20/E2ユーザインターフェースケーブルを接続してもらう。

(7) ex\_boardinfo2() ： E1/E20/E2ユーザインターフェースケーブルの接続(ターゲット電源ON)を確認する

(8) ex\_socunitinfo2() unit\_type=EX\_UNIT\_CG

(9) ex\_socunitinfo2() unit\_type=EX\_UNIT\_INITSTOPDBG★

※デバイスファイルのEFIタグでfunc:0x0103＜エミュレーション機能情報3＞の

bit3(Initial Stop Debug)が1の場合、初期停止・スタンバイモードデバッグ環境の有効/無効を選択することが可能である。呼び出しをしなかった場合は、無効が選択される。

詳細は、ex\_socunitinfo2の[Description]を参照する事。

(10) ex\_socunitinfo2() unit\_type=EX\_UNIT\_LPDOPT

※ホットプラグインの場合、デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1であってもE1エミュレータによる端子リセット制御を行わないため、端子リセットディレイ対策は行う必要がない。詳細は、ex\_socunitinfo2の[Description]を参照する事。

(11) ex\_socunitinfo2() unit\_type= EX\_UNIT\_FID

※DFファイル内EFIタグ「エミュレーション機能情報1」のbit18(CodeFlash-ID)、bit19(DataFlash-ID)の各情報が‘1’の場合は、本コマンドの発行が必要である。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、ex\_socunit\_info2()unit\_type=EX\_UNIT\_RSUのunit\_info[4](unit\_info[8])、あるいは起動手順完了後(注1)に本コマンドを参照指定で発行することで確認できる。

(12) ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU ： ホットプラグインRUN状態で接続する。

※ホットプラグインRUN中(接続～強制ブレークするまで)は、通常動作と異なるEXEC関数もあるので注意

(13) ex\_socunitinfo2() unit\_type=EX\_UNIT\_COREPESEL

デバイスファイルで有効なPE数分だけ(13)を繰り返し、無効なPEがないか確認する。

(14) ex\_getenv()

(15) ex\_rh\_getenv()

★のI/Fの呼び出しは必須ではない。呼び出さなくても起動する。

注1：起動手順完了後とは、ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU発行後のことである。

(ICUM有効モードでかつC&R認証が必要な場合を除く)

＜ホットプラグイン起動時の状態について＞

ホットプラグイン起動後のユーザプログラム実行状態は”ホットプラグインRUN”となり、通常起動のユーザプログラム実行状態と異なる。

＜G4世代MCU 起動時に必要なEXEC関数の手順(通常起動)＞

(1) ex\_getemulist()★

(2) ex\_setemulator()★

(3) ex\_boardinfo2()

＜ターゲット接続時の注意事項＞

ターゲット電源OFF時は、電源ON後に再度ex\_boardinfo2()を呼び出す必要がある

(4) ex\_setcallback\_sts()★

＜注意事項＞

呼び出す必要があるのは、非同期通信I/Fを使用する場合のみ。

(5) ex\_extpower()★

＜注意事項＞

E2エミュレータ/IE850Aエミュレータ以外では呼び出し禁止。E2エミュレータ/IE850Aエミュレータでは呼び出さなくても起動は可能だが、E2エミュレータ/IE850Aエミュレータサポート時は本I/Fを呼び出すようにすること。

(6) ex\_initexec()

(7) ex\_socunitinfo2() unit\_type=EX\_UNIT\_CG

(8) ex\_socunitinfo2() unit\_type=EX\_UNIT\_INITSTOPDBG★

※デバイスファイルのEFIタグでfunc:0x0103＜エミュレーション機能情報3＞の

bit3(Initial Stop Debug)が1の場合、初期停止・スタンバイモードデバッグ環境の有効/無効を選択することが可能である。呼び出しをしなかった場合は、無効が選択される。

詳細は、ex\_socunitinfo2の[Description]を参照する事。

(9) ex\_socunitinfo2() unit\_type=EX\_UNIT\_LPDOPT

※デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合、端子リセットディレイ対策の選択を行う必要がある。詳細は、ex\_socunitinfo2の[Description]を参照する事。

(10) ex\_socunitinfo2() unit\_type= EX\_UNIT\_CMFID

※DFファイルのEFI部のCustomer ID認証情報(func=0x0115、index=0x0001)、およびDFファイルのEFI部のData Flash ID認証情報(func=0x0116、Index=0x0001)を参照し、必要な回数だけ繰り返し発行する。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、起動手順完了後に本コマンドを参照指定で発行することで確認できる。

(11) ex\_socunitinfo2() unit\_type= EX\_UNIT\_OPTID★

※DFファイルのEFI部の任意ID認証情報(func=0x0117、index=0x0001)を参照し、必要な回数だけ繰り返し発行する。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、起動手順完了後(注1)に本コマンドを参照指定で発行することで確認できる。

(12) ex\_socunitinfo2() unit\_type=EX\_UNIT\_SVR

　　※DFのEFIタグでfunc:0x0103<エミュレーション機能情報3>のbit00(SVR)が1の場合に呼び出す。

(13) ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU

　必要であれば、(8)～(13)を繰り返す

(14) ex\_socunitinfo2() unit\_type=EX\_UNIT\_COREPESEL

デバイスファイルで有効なPE数分だけ(14)を繰り返し、無効なPEがないか確認する。

(15) ex\_pincont()

(16) ex\_getenv()

(17) ex\_rh\_getenv()

★のI/Fの呼び出しは必須ではない。呼び出さなくても起動する。

注1：起動手順完了と後は、ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU発行後のことである。

(ICUM有効モードでかつC&R認証が必要な場合を除く)

＜G4世代MCU ホットプラグイン起動の手順(OCDのみ)＞

(1) ex\_getemulist()★

(2) ex\_setemulator()★

(3) ex\_boardinfo2() ： ホットプラグインアダプタにE1/E20/E2ユーザインターフェースケーブルが未接続

(ステータス情報からターゲット電源OFF)を確認する

(4) ex\_setcallback\_sts()★

※呼び出す必要があるのは、非同期通信I/Fを使用する場合のみ。

(5) ex\_extpower()★ ： ホットプラグイン起動指定で呼び出す

＜注意事項＞

E2エミュレータ/IE850Aエミュレータ以外では呼び出し禁止。E2エミュレータ/IE850Aエミュレータでは呼び出さなくても起動は可能だが、E2エミュレータ/IE850Aエミュレータサポート時は本I/Fを呼び出すようにすること。

(6) ex\_initexec() ： ホットプラグイン起動指定で呼び出す

ユーザにE1/E20/E2ユーザインターフェースケーブルを接続してもらう。

(7) ex\_boardinfo2() ： E1/E20/E2ユーザインターフェースケーブルの接続(ターゲット電源ON)を確認する

(8) ex\_socunitinfo2() unit\_type=EX\_UNIT\_CG

(9) ex\_socunitinfo2() unit\_type=EX\_UNIT\_INITSTOPDBG★

※デバイスファイルのEFIタグでfunc:0x0103＜エミュレーション機能情報3＞の

bit3(Initial Stop Debug)が1の場合、初期停止・スタンバイモードデバッグ環境の有効/無効を選択することが可能である。呼び出しをしなかった場合は、無効が選択される。

詳細は、ex\_socunitinfo2の[Description]を参照する事。

(10) ex\_socunitinfo2() unit\_type=EX\_UNIT\_LPDOPT

※ホットプラグインの場合、デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1であってもE1エミュレータによる端子リセット制御を行わないため、端子リセットディレイ対策は行う必要がない。詳細は、ex\_socunitinfo2の[Description]を参照する事。

(11) ex\_socunitinfo2() unit\_type= EX\_UNIT\_CMFID

※DFファイルのEFI部のCustomer ID認証情報(func=0x0115、index=0x0001)、およびDFファイルのEFI部のData Flash ID認証情報(func=0x0116、Index=0x0001)を参照し、必要な回数だけ繰り返し発行する。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、起動手順完了後に本コマンドを参照指定で発行することで確認できる。

(12) ex\_socunitinfo2() unit\_type= EX\_UNIT\_OPTID★

※DFファイルのEFI部の任意ID認証情報(func=0x0117、index=0x0001)を参照し、必要な回数だけ繰り返し発行する。

※本コマンド発行時点では、各セキュリティIDの認証結果は返らない。認証結果は、起動手順完了後(注1)に本コマンドを参照指定で発行することで確認できる。

(13) ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU ： ホットプラグインRUN状態で接続する。

※ホットプラグインRUN中(接続～強制ブレークするまで)は、通常動作と異なるEXEC関数もあるので注意

(14) ex\_socunitinfo2() unit\_type=EX\_UNIT\_COREPESEL

デバイスファイルで有効なPE数分だけ(14)を繰り返し、無効なPEがないか確認する。

(15) ex\_getenv()

(16) ex\_rh\_getenv()

★のI/Fの呼び出しは必須ではない。呼び出さなくても起動する。

注1：起動手順完了と後は、ex\_socunitinfo2() unit\_type=EX\_UNIT\_RSU発行後のことである。

(ICUM有効モードでかつC&R認証が必要な場合を除く)

＜ホットプラグイン起動時の状態について＞

ホットプラグイン起動後のユーザプログラム実行状態は”ホットプラグインRUN”となり、通常起動のユーザプログラム実行状態と異なる。

### 終了

終了はex\_exit()関数を呼出すことでEXECとの通信をクローズする。

終了後に再起動する場合は、再び2.2.1起動の手順を行うこと。

## CPUリセットが発生する関数

表 2‑5 CPUリセット関連表

|  |  |
| --- | --- |
| リセット発生操作 | 対象 |
| ex\_socunitinfo2()を以下のunit\_typeを指定し、設定(EX\_SET)で呼び出した場合 |  |
| unit\_type=EX\_UNIT\_RSU | E1、E20、E2、IE850A ※1 |
| unit\_type=EX\_UNIT\_DCUSEC | E1、E20、E2、IE850A |
| unit\_type=EX\_UNIT\_FLMKOP | E1、E20、E2、IE850  IE850A |
| unit\_type=EX\_UNIT\_EXFLMKOP | E1、E20、E2、IE850A |

※1　ホットプラグイン接続時にはCPUリセットが発生しない

## 低消費電力モード時の動作仕様

各低消費電力モード時のEXEC動作仕様を以下に示す。

(1)ソフトウェアスタンバイ(電源遮断)モード

ソフトウェアスタンバイ(電源遮断)モードのエミュレーションは行えない。

このモードに遷移させて、その後電源をONした場合、エミュレータの再接続処理が必要である。

(2)HALTモード

EXECではHALTモード時も通常状態と同様の動作を行う。

そのため、HALTモード時にデバイスメモリやシステムレジスタなど一旦ブレークしてアクセスするI/Fを発行した場合、HALTモードは解除される。

(3)STOPモード

初期停止・スタンバイモードデバッグ環境が無効時は、STOPモード中も通常状態と同様の動作を行う。

ただし、STOPモード時にもEXECから制御できるようにユーザプログラム上でWUFMSK0[31]=0に設定しておくこと。

STOPモード時にデバイスメモリやシステムレジスタなどへアクセスするI/Fを発行した場合、STOPモードは解除される。

初期停止・スタンバイモードデバッグ環境が有効時は、EXECからデバッグモジュールに対する制御ができないため、本モード中にデバッグ機能に関するEXEC I/Fを呼び出した場合はエラー応答する。

(4)DEEP STOPモード

初期停止・スタンバイモードデバッグ環境が無効時、DEEP STOPモード中は通常状態とは異なる動作を行う。通常動作ではIso領域の電源供給が停止するがエミュレーション時はIso領域の電源供給は継続される。

また、STOPモード中と同様にEXECから制御できるようにユーザプログラム上でWUFMSK0[31]=0に設定しておくこと。

DEEP STOPモード時にデバイスメモリやシステムレジスタなどへアクセスするI/Fを発行した場合、DEEP STOPモードは解除される。

初期停止・スタンバイモードデバッグ環境が有効時は、EXECからデバッグモジュールに対する制御ができないため、本モード中にデバッグ機能に関するEXEC I/Fを呼び出した場合はエラー応答する。

(5)モジュールスタンバイモード

モジュールスタンバイモード状態時でも、対象の周辺モジュールのレジスタなどへアクセスすることは可能である。そのため、EXECはモジュールスタンバイモード状態でも通常時と同様の動作を行う。

(6)Cyclic RUNモード

初期停止・スタンバイモードデバッグ環境が無効時は、Cyclic RUNモード中にブレークを伴うI/Fを発行するとエミュレーションが不可能な状態に陥り、再接続処理が必要である。

初期停止・スタンバイモードデバッグ環境が有効時は、Cyclic RUNモード中でもアクセス可能な領域に限り、EXECから制御が可能である。

(7)Cyclic STOPモード

初期停止・スタンバイモードデバッグ環境が無効時は、Cyclic STOPモード中にブレークを伴うI/Fを発行するとエミュレーションが不可な状態に陥り、再接続処理が必要である。

初期停止・スタンバイモードデバッグ環境が有効時は、EXECからデバッグモジュールに対する制御ができないため、本モード中にデバッグ機能に関するEXEC I/Fを呼び出した場合はエラー応答する。

## ターゲット電源に関する動作仕様

ターゲット電源OFFの間は以下の関数以外のEXEC関数は使えなくなる。これは、ターゲット電源OFFで、デバッグ対象のCPUの電源が切れてしまい通信そのものが切断さるためである。

　ex\_exit、ex\_getstatus、ex\_boardinfo2

ターゲット電源OFFからONへの復帰をサポートするのはD1xのIE850 PODのみである。

EXECはターゲット電源OFFからONになるのを待ち、ONになったらデバッグが再開可能となる。

~~EXECはターゲット電源OFFからONになるのを待ち、ONになったらデバッグ対象のCPUと通信を再接続し、デバッグを続けられるように電源OFFで初期化されたデバッグ機能の再設定処理を行う。~~

ターゲット電源に関するEXEC動作仕様を以下に示す。

＜ターゲット電源OFF＞

・ex\_getstatusを実行するとexestatステータス情報のターゲット電源の状態が、電源OFFとなる。

・ターゲット電源OFF前がブレーク中の場合、ユーザプログラムブレーク状態からユーザプログラム実行状態に変わる。ターゲット電源OFF前がユーザプログラム実行状態の場合は実行状態を維持する。(電源ON時のユーザプログラム実行状態でex\_getstatusを実行するとPC値としてFFFFFFFFhを返すが、電源OFF時のユーザプログラム実行状態ではPC値として1を返す)

・ターゲット電源OFF前がユーザプログラム実行中でトレース起動中だった場合、ターゲット電源がOFFになってもトレース起動中のステータスは維持する。トレース停止中だった場合も、そのステータスを維持する。(CS+がこのステータスを見てトレース取得しないようにするための対策)

・ターゲット電源OFFの間にEXEC関数を発行した場合はEX\_SE\_TRGTOFFを返す。  
(ex\_exit、ex\_getstatus、ex\_boardinfo2は除く)

・ターゲット電源OFF状態でex\_exitによりEXECを終了した場合、ターゲット電源ONにした後は再接続が可能になる。

＜ターゲット電源OFF→ON＞

・EXECがCPUのデバッグを再開する。

・ターゲット電源OFF前がユーザプログラム実行中の場合、ユーザプログラム実行を再開する。  
ターゲット電源OFF前がブレーク中の場合、要因なしでブレークとなる。

・exestatステータス情報のターゲット電源の状態が、電源ONとなる。

~~IE850もPODのデバッグチップの電源を常にターゲットから供給している機種は上記と同様の仕様となる。~~

~~ターゲット電源OFFになった場合、内蔵トレースの取得データやシーケンシャルの成立条件などは消えてしまう。~~

D1xのIE850 PODではターゲット電源をOFFにしても、実際にはIE850本体から電源を供給し続けるため、デバッグリソースは保持される。このため、電源OFF前の各デバッグ機能の設定は保持され、シーケンシャルイベントの成立条件、埋め込み方式のソフトウェアブレークなども保持され、トレース機能も継続される。

本開発バージョンではD1xのIE850 PODのみサポートする。その他のIE850 PODやOCDではターゲット電源をOFFにすることを禁止とする。

D1xのIE850 PODには、1種類のコア電源(REG0VCC)を制御するPODと2種類のコア電源(REG0VCC、REG1VCC)を制御するPODがある。

ユーザプログラム実行中にREG0VCCのコア電源がOFF→ONした場合、POC0(リセットレベル1)のエミュレーションをするため、同じリセットレベル1である強制リセットをEXECが発行する。REG1VCCのコア電源がOFF→ONした場合、POC1(リセットレベル2)のエミュレーションをするため、同じリセットレベル2であるPWRGDによるリセットをIE850 POD上のハードウェアが発行する。なお、REG1VCCのコア電源がOFF中にREG0VCCもOFFした場合(同時に2種類のコア電源がOFFした場合も含む)は、リセットレベル1である強制リセットをEXECが発行する。

ブレーク中は、REG1VCCのコア電源がOFF→ONした場合でも、REG0VCCのコア電源と同様にリセットレベル1である強制リセットをEXECが発行する。

[デバッガ関係者向け注意事項]

ターゲット電源OFF時にex\_exitコマンドを使用可能としているが、電源を再ONしてからのex\_exitコマンド使用を推奨する。電源OFF状態で切断すると、通常の電源OFFルーチンとは異なり、ソフトウェアブレークの書き戻し等が行われないため。

## 仮想化支援機能に関する動作仕様

[RH850G3]

V850E3v5アーキテクチャで定義されたプロセッサモデルの資源管理方式の内、仮想マシンに対応した仮想化支援機能のデバッグをサポートする。

以下の機能で仮想化支援機能をサポートする。

・ハードウェア条件設定機能

ex\_getstatusの拡張でNM/VMの状態を返す。

ex\_socunitinfo2のunit\_type=EX\_UNIT\_COREPESELの拡張でコンテキストを選択する。

ex\_socunitinfo2のunit\_type=EX\_UNIT\_VIRTUALINFOの追加で仮想化支援機能の使用有無を選択する。

・レジスタ操作機能

ex\_socunitinfo2のunit\_type=EX\_UNIT\_COREPESELで選択したモードのコンテキストを操作対象とする。

・イベント条件機能

仮想化支援機能対応有無もしくは仮想化支援機能使用有無に関わらず、NM/VM/HTのいずれのモードでもイベント検出可能な設定にする。

・アナライザ機能

(1)トレース

仮想化支援機能対応有無もしくは仮想化支援機能使用有無に関わらず、NM/VM/HTのいずれのモードでもトレース可能な設定にする。

また、トレースされたデータへのNM/VM/HT状態の反映はトレースメッセージに依存する。トレースメッセージのフォーマットは「G3世代 トレース・コントロール・ユニット(TCU)機能仕様書」を参照すること。

(2)タイマ

仮想化支援機能対応有無もしくは仮想化支援機能使用有無に関わらず、NM/VM/HTのいずれのモードでも時間計測可能な設定にする。

・非同期通信機能

コールバック用関数(callback\_sts)の拡張でNM/VMの状態を返す。

[仮想マシン対応時の注意点]

シングルデバッグスコープでは、ブレーク時はVMモードで実行されていても必ずNMモードに切り替わる。この動作を踏まえた上で、ex\_gatstatusやコールバックのステータス機能、トレース機能の情報を使用すること。

nVM/nHTおよびマルチデバッグスコープの対応を行う時点で、サポート方法が変更となる。

(仮想化支援機能に対応した新規関数の追加による既存関数の置き換えを想定しておく必要がある)

また、仮想記憶機能等のアドレス変換には対応せず、すべて物理アドレスとして処理する。

[RH850G4]

RH850 version 2.1 アーキテクチャで追加されたCPU仮想化支援機能のデバッグをサポートする。

仮想化支援機能をサポートしているかどうかはデバイスファイルのEFIタグでfunc:0x0103＜エミュレーション機能情報3＞のbit1(Virtualization(G4))で判断できる(1：仮想化支援機能をサポート)。

ハイパーバイザソフトウェアが動作するモードをホストモード、仮想マシン上でソフトウェアが動作するモードをゲストモードと呼び、総称してCPU動作モードと呼ぶ。一方で仮想化支援機能を無効化した状態を従来モードと呼ぶ。

以下の機能で仮想化支援機能をサポートする。

・ハードウェア条件設定機能

ex\_getstatusの拡張で(ホストモードor従来モード)/ゲストモードの状態、ゲスト・パーティション識別子(GPID)を返す。

ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELの追加でデバッグ対象のCPU動作モードを選択する。

また、ブレーク時にカレントコアのブレークPC位置をデバッグ対象のCPU動作モードまで滑らせるオプションとステップ実行時にデバッグ対象外のCPU動作モードで停止しないオプションを指定できる。

・レジスタ操作機能

I/F呼び出し時にホストモードであれば、ホストモードのコンテキストレジスタに対してアクセスを行う。ゲストモードであれば、その時点のゲストモードのコンテキストレジスタに対してアクセスを行う。また、レジスタアクセスを行う際は、常にHV権限でアクセスするため、すべてのレジスタに対してアクセスが可能となる。また、ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELでデバッグ対象外に指定したCPU動作モードの状態でレジスタアクセスした場合、エラー応答する。

・イベント条件機能

仮想化支援機能有効時は、ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELでデバッグ対象に選択したCPU動作モードでイベント検出可能な設定にする。

・トレース機能、外部ソフトトレース単線出力機能

仮想化支援機能有効時は、ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELでデバッグ対象に選択したCPU動作モードのトレース・データのみが取得できる。デバッグ非対象にしたCPU動作モードのトレース・データは取得しない。トレース機能に関しては、CPU動作モードの遷移(ゲスト←→ホスト)のトレース・データの出力有無をex\_rh\_trceventsel2で選択できる。

・パフォーマンス機能(TMU\_CLKタイマを含む)

仮想化支援機能有効時は、ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELでデバッグ対象に選択したCPU動作モードで実行されている区間の現象のみが計測される。

RUN-BREAKタイマ、TMU\_TCKタイマはデバッグクロックによる計測で常時計測されるため、仮想化支援機能の有効/無効に関わらず、常時計測される。

・非同期通信機能

コールバック用関数(callback\_sts)の拡張で(ホストモードor従来モード)/ゲストモードの状態、ゲスト・パーティション識別子(GPID)を返す。

[仮想化支援機能対応時の注意点]

1. G4世代ではex\_socunitinfo2のunit\_type=EX\_UNIT\_VIRTUALINFOによる仮想化支援機能の使用有無は対応せず、常に有効状態と同様に扱う。デバイスの仮想化支援機能が無効(HVCFG.HVE=0、従来モード)の場合は、状態としてはホストモードと同等となる。また、この場合ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELによる設定は影響を受けずに動作する。
2. デバッグ対象としてCPU動作モードを指定しても、イベント検出やトレース取得、ハードウェアブレークができない場合がある。DBGENレジスタ(ユーザ公開予定)の設定によってはイベント検出やトレース取得ができなくなる。DBGENレジスタはユーザ資源であり、ユーザが意図してレジスタ設定を行っているため、デバッガからDBGENレジスタの制御は行わない。
3. G3世代ではex\_socunitinfo2のunit\_type=COREPESELでレジスタアクセスするコンテキストを指定したが、G4世代ではI/F呼び出し時のCPU動作モードのコンテキストレジスタがアクセス対象となり、G3世代とは異なる。

## 非同期デバッグに関する動作仕様

マルチコアデバイスにおいて、非同期デバッグモードをサポートする。以下の機能で従来の同期デバッグモードと非同期デバッグモードの切り替えを行う。

　ex\_socunitinfo2 unit\_type = EX\_UNIT\_ASYNCPEOPT

同期デバッグモードにおいては複数コア間でユーザプログラム実行開始およびブレークのタイミングは同期するが、非同期デバッグモードの場合、ユーザプログラム実行開始およびブレークをそれぞれのコアを選択した状態で別々に指定することができる(同時に指定することはできない)。

非同期デバッグモードにおいては、特定のコアをユーザプログラム実行させた状態で別のコアに注目したデバッグをすることを目的としている。ここで注目してデバッグを行うコアを「デバッグ対象コア」と定義する。一方、デバッグ対象コアをデバッグする間ユーザプログラム実行させておくコアを「デバッグ非対象コア」と定義する。非同期デバッグにおいて、マルチコアのすべてのコアで従来の同期デバッグで行っていたデバッグ機能をサポートすることはできないため、デバッグ非対象コアのデバッグ機能を必要最低限のものだけサポートする。デバッグ対象コアにおいては同期デバッグでサポートするデバッグ機能に近づけたデバッグ機能をサポートするが、ソフトウェアブレークは命令埋め込み方式であるため、デバッグ対象コア、デバッグ非対象コアともにサポートできない。各コアがサポートするデバッグ機能を表 2‑6に示す。

表 2‑6 非同期デバッグモードにおけるサポート機能Support functions in asynchronous debug mode

|  |  |  |  |
| --- | --- | --- | --- |
| 機能 | I/F関数 | デバッグ対象コアCore to be debugged | デバッグ非対象コアDebug non-target core |
| 実行 | ex\_run | ○ | ○ |
| ブレーク | ex\_stop | ○ | ○ |
| ステップ | ex\_step | ○ | ○ |
| リセット | ex\_reset | ○※1 | ○※1 |
| リセット  +実行 | ex\_resetrun | ○※5 | ○※5 |
| メモリ | ex\_memwrite | ○※2 | ○※2 |
| ex\_memread | ○ | ○ |
| ex\_memfill | ○ | ○ |
| ex\_memcopy | ○※3 | ○※3 |
| レジスタ | ex\_regwrite | ○ | ○ |
| ex\_regread | ○ | ○ |
| イベント  ブレーク | ex\_rh\_evncond | ○ | ○ |
| ex\_rh\_evnintgcond | ○ | ○ |
| ex\_hwbrkevent | ○ | ○ |
| SWブレーク | ex\_swbrkcond | **×**(EX\_SE\_ASYNCMODE\_ENB) | **×**(EX\_SE\_ASYNCMODE\_ENB) |
| トレース | ex\_trcsw | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcrun | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcmode | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcclear | ○ | ○ |
| ex\_trcevent | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcdelay | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_rh\_trceventsel2 | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_rh\_trcbufcont | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcrawmeminfo | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_trcrawmemread | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_rh\_trchwevncopeinfo | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| ex\_rh\_trctune | ○ | **×**(EX\_SE\_ASYNCMODE\_SELPEDSB) |
| 外部ソフトトレース単線出力  External soft trace single wire output | ex\_rh\_sfttrcsw | ○ | ○ |
| ex\_rh\_sfttrcmode | ○ | ○ |
| ex\_rh\_sfttrcevent | ○ | ○ |
| ex\_rh\_sfttrcfilter | ○ | ○ |
| ex\_rh\_sfttrctune | ○ | ○ |
| ex\_rh\_sfttrcrawmemmode | ○ | ○ |
| ex\_rh\_sfttrcrawmeminfo | ○ | ○ |
| ex\_rh\_sfttrcrawmemread | ○ | ○ |
| ハードウェア設定 | ex\_setenv | ○※3 | ○※3 |
| ex\_socunitinfo2 | ○※4 | ○※4 |
| タイマ | ex\_timesw | ○ | ○ |
| ex\_timeclear | ○ | ○ |
| ex\_timeread2 | ○ | ○ |
| ex\_rh\_timeread | ○ | ○ |
| ex\_timeevent2 | ○ | ○ |
| ex\_rh\_timeevent | ○ | ○ |
| パフォーマンス | ex\_rh\_performancesw | ○ | ○ |
| ex\_rh\_performanceclear | ○ | ○ |
| ex\_rh\_performanceread | ○ | ○ |
| ex\_rh\_performanceset | ○ | ○ |

○：同期デバッグ時と同等で使用可能、×：使用不可、()内は返るエラーを示す。

※1：リセットは全コア同時(対象コアも非対象コアも同期してRESETが入る)

※2：コードフラッシュ、データフラッシュへの書き込みは全コアがブレーク時のみ使用可能

全コアブレークしていない場合はエラー(EX\_SE\_ANOTHERPERUN)が返る

※3：全コアブレークしている状態の時のみ使用可能

全コアブレークしていない場合はエラー(EX\_SE\_ANOTHERPERUN)が返る

※4：unit\_typeにより全コアブレーク時のみ可能な機能あり(詳細はex\_socunitinfo2参照)

※5：リセットは全コア同時になるため、実行も全コア同時に行う

[サポートしない機能について]

各コア個別で設定されるトレース関連の条件設定は、非同期デバッグモードに遷移するとデバッグ非対象コアではI/F関数を呼び出すことができなくなり、設定も一時的に無効となるが、その設定は内部的に保持される。再び、同期デバッグモードに遷移すると、内部的の保持していた設定が復帰される。

これに該当するI/Fは以下の通り。

|  |  |  |
| --- | --- | --- |
| ex\_trcsw | ex\_trcevent | ex\_trcdelay |
| ex\_rh\_trceventsel2 | ex\_rh\_trcbufcont | ex\_rh\_trctune |

※ex\_swbrkcondはデバッグ非対象コアだけでなく、デバッグ対象コアでもサポートしないが、同様に、非同期デバッグモードに遷移するとブレークポイントは無効になるが、一時的に内部で保持し、同期デバッグモードに遷移すると、ブレークポイントが復元される。

上記とは別に、全コアで共通に設定される条件設定等は、各コアから設定変更が可能になるためデバッグ非対象コアから変更しなくてもデバッグ対象コアから変更される可能性があるため、非同期デバッグモード選択時は、一時的に内部で保持は行わない。これに該当するI/Fは以下の通り。

　ex\_trcmode

[非同期デバッグモードにおける制限事項]

非同期デバッグモードでは制限となる事項を以下に示す。

(1)ex\_setenvのse.pev\_modeで指定できる周辺チップの動作指定で「周辺マクロを停止する」に設定しても周辺マクロは停止しない。設定されてもエラーは返さない。

(2)操作対象外のPEがユーザプログラム実行状態で特定シーケンスによるアクセス中に、操作対象のPEによるI/O領域へのアクセスが発生した場合、操作対象外のPEで実行中の特定シーケンスが破壊される可能性がある。

(3)ECCに関する制限

①操作対象外のPEがユーザプログラム実行状態で操作対象のPEからコードフラッシュ、データフラッシュ、ECC付き内蔵I/O領域へのメモリアクセスを行うと、一時的にECCのOFF制御を行うため、ユーザプログラム実行中の操作対象外のPEでECCエラーを検出できない可能性がある。

②ECC制御レジスタを破壊する可能性がある。

例.ブレーク状態でコードフラッシュを読み出す場合

1. ECC制御レジスタの値を退避

2. ECC OFFのためのECC制御レジスタを設定

3. コードフラッシュを読み出す

4. 上記1で退避したECC制御レジスタの値に書き戻し

コードフラッシュを読み出す場合は上記手順で行うため、上記のメモリ操作を行っていないPEがRUN状態で1～3の期間にECC制御レジスタを操作した場合、ユーザ設定値と異なるレジスタ値を④で設定する場合がある。

(4)トレース出力先に内蔵トレースを選択しているとき、トレース・データの記録動作条件のフルブレーク(EX\_FULL\_BRK)と非同期デバッグモードの組み合わせは設定することができない。この組み合わせに設定したとき、各I/Fでエラーを返す。エラーの詳細はex\_socunitinfo2(EX\_UNIT\_ASYNCPEOPT)とex\_trcmodeを参照。

(5)ソフトウェアブレークの設定/参照/削除ができない。ex\_swbrkcondを呼び出した場合はエラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

(6)いずれかのコアがブレークしている場合、内部リセットおよび外部リセットはマスクされる。ex\_pincontでリセットをマスクしない設定にしている場合でも、いずれかのコアがブレークしている場合はリセットがマスクされる。

(7)いずれかのコアがブレークしている場合、ウォッチドッグタイマのカウント動作が停止する。

(8)スレーブイベントおよびスレーブトレースはサポートしない。非同期デバッグモードの時に、スレーブ関連のI/Fを呼び出したときにもエラー応答は行わず、設定は行われる。ただし、スレーブイベントおよびスレーブトレースの動作は行わない。(非同期デバッグモード時の設定は、同期デバッグモードに切り替えたときに反映される。)

(9)外部ソフトトレース単線出力機能はサポートしない。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

同期デバッグモードと非同期デバッグモードの切り替えは、停止(初期停止・Cyclic Disable)中に遷移しているコアが存在する状態では行えない。

[デバッガ関係者向け注意事項]

表 2‑6で使用不可(×)となっている機能のI/F関数を呼び出すとエラーを返す。

ユーザプログラム実行およびブレーク状態のステータス検出方法に関して、同期デバッグモードでは選択コアのステータス取得によってデバイス全体の状態を知ることができるが、非同期デバッグモードでは各コア単位でステータスを取得する必要があるので注意が必要。

## E2エミュレータ拡張機能に関する動作仕様Operation specifications for E2 emulator extension function

E2エミュレータではE1/E20相当のデバッグ機能に加え、外部トリガ入出力やCANモニタ、外部ソフトトレース単線出力などの拡張機能をサポートする(外部ソフトトレース単線出力については単体機能としても公開する)。拡張機能はある事象を検出し、ユーザプログラム実行を停止したり、ユーザプログラムの停止を検出して、拡張機能を動作させたりすることができる。ただし、マルチコアデバイスにおいて、どのコアのユーザプログラム実行/停止に結び付けるかを選択できないため、非同期デバッグモードとの併用は不可とする。非同期デバッグモード中に拡張機能を有効化しようとするとRFW I/F関数でエラーが返る(詳細はE2エミュレータ 拡張機能部 インターフェース関数仕様書参照)。同様に、拡張機能が有効な状態で、非同期デバッグモードに設定しようとした場合、エラーEX\_SE\_CANTSET\_ASPONが返る。

In addition to the E1 / E20 debugging function, the E2 emulator supports extended functions such as external trigger input / output, CAN monitor, and external soft trace single line output (external soft trace single line output is also disclosed as a single function). The extended function can detect a certain event and stop the execution of the user program, or can detect the stop of the user program and operate the extended function. However, in a multi-core device, since it cannot be selected which user program execution / stop is linked to, it cannot be used in combination with the asynchronous debug mode. An error is returned in the RFW I / F function when the extension function is enabled in the asynchronous debug mode (refer to the E2 emulator extension function interface function specifications for details). Similarly, error EX\_SE\_CANTSET\_ASPON is returned if an attempt is made to set asynchronous debug mode while the extended function is enabled.

また、拡張機能として外部ソフトトレース単線出力を使用する場合、ターゲットI/Fがソフトトレース・データを送信する専用の端子動作となるため、ターゲットマイコンにアクセスする既存機能は使用できない。外部ソフトトレース単線出力を使用しない場合でも、FPGAがターゲットステータスを監視するため、LPD通信はH/Wが占有し、F/Wからターゲットマイコンにアクセスする機能は使用できない。ターゲットマイコンにアクセスするI/F関数を呼び出した場合、エラーEX\_SE\_CANTCALL\_ASPRUNNINGが返る。ただし、強制ブレークとステータス取得などは例外として使用可能とする。拡張機能有効状態でユーザプログラム実行中のI/F関数の呼び出し可否を以下の表に示す。

In addition, when using external soft trace single-wire output as an extended function, the target I / F functions as a dedicated terminal that transmits soft trace data, so existing functions that access the target microcomputer cannot be used. Even when the external soft trace single line output is not used, because the FPGA monitors the target status, LPD communication is occupied by H / W, and the function to access the target microcomputer from F / W cannot be used. When an I / F function that accesses the target microcomputer is called, the error EX\_SE\_CANTCALL\_ASPRUNNING is returned. However, forced break and status acquisition can be used as exceptions. The following table shows whether the I / F function can be called during execution of the user program with the extended function enabled.

表 2‑7 拡張機能有効時のRUN中のI/F関数呼び出し可否Whether I / F function can be called during RUN when extended function is enabled

|  |  |  |  |
| --- | --- | --- | --- |
| I/F関数名function name | RUN中の呼び出し可否Callability during RUN | | 備考Remarks |
| 拡張機能無効Extension disabled | 拡張機能有効Extended function enabled |
| ex\_exit | ○ | ○ | 呼び出し可。USB切断時などの非常事態に呼び出される想定。 |
| ex\_initexec | × | × | 元々呼び出し不可 |
| ex\_reset | ○ | × | ターゲットアクセスできないため不可 |
| ex\_setemulator | × | × | 元々呼び出し不可 |
| ex\_getemulist | × | × | 元々呼び出し不可 |
| ex\_extpower | ○ | ○ | ターゲットアクセスないため可 |
| ex\_getenv | ○ | ○ | ターゲットアクセスないため可Possible because there is no target access |
| ex\_rh\_getenv | ○ | × | ターゲットアクセスできないため不可Not possible because target cannot be accessed |
| ex\_getstatus | ○ | ○ | 呼び出し可。RUN/BRK情報は返す |
| ex\_setcallback\_sts | ○ | ○ | ターゲットアクセスないため可 |
| ex\_boardinfo2 | ○ | × | ターゲットアクセスできないため不可 |
| ex\_setenv | × | × | 元々呼び出し不可 |
| ex\_pincont | ○ | × | ターゲットアクセスできないため不可 |
| ex\_m\_montimeout | ○ | ○ | ターゲットアクセスないため可 |
| ex\_socunitinfo2 | ※ | ※ | 拡張機能有効/無効に依存しない |
| ex\_memread | ○ | × | ターゲットアクセスできないため不可 |
| ex\_memwrite | ○ | × | ターゲットアクセスできないため不可 |
| ex\_memfill | ○ | × | ターゲットアクセスできないため不可 |
| ex\_memcopy | × | × | 元々呼び出し不可 |
| ex\_regread | ○ | × | ターゲットアクセスできないため不可 |
| ex\_regwrite | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_evncond | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_evnintgcond | ○ | × | ターゲットアクセスできないため不可 |
| ex\_run | × | × | 元々呼び出し不可 |
| ex\_resetrun | × | × | 元々呼び出し不可 |
| ex\_step | × | × | 元々呼び出し不可 |
| ex\_stop | ○ | ○ | 呼び出し可 |
| ex\_hwbrkevent | ○ | × | ターゲットアクセスできないため不可 |
| ex\_swbrkcond | × | × | 元々呼び出し不可 |
| ex\_trcsw | × | × | 元々呼び出し不可Not callable originally |
| ex\_trcrun | ○ | × | ターゲットアクセスできないため不可Not possible because target cannot be accessed |
| ex\_trcmode | ○ | × | ターゲットアクセスできないため不可 |
| ex\_trcclear | ○ | × | ターゲットアクセスできないため不可 |
| ex\_trcevent | ○ | × | ターゲットアクセスできないため不可 |
| ex\_trcdelay | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_trceventsel2 | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_trcbufcont | ○ | × | ターゲットアクセスできないため不可 |
| ex\_trcrawmeminfo | ○ | × | ターゲットアクセスできないため不可 |
| ex\_trcrawmemread | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_trchwevncopeinfo | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_trctune | × | × | 元々呼び出し不可 |
| ex\_rh\_sfttrcsw | ○ | × | ターゲットアクセスできないため不可  Not possible because target cannot be accessed |
| ex\_rh\_sfttrcmode | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_sfttrcfilter | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_sfttrctune | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_sfttrcrawmemmode | × | × | 有効時は呼び出し不可  無効時は拡張機能の制限により不可  Cannot be called when enabled When disabled, it is not possible due to the limitations of the extension |
| ex\_rh\_sfttrcrawmeminfo | × | × | 有効時は呼び出し不可  無効時は拡張機能の制限により不可 |
| ex\_rh\_sfttrcrawmemread | × | × | 有効時は呼び出し不可  無効時は拡張機能の制限により不可 |
| ex\_timesw | × | × | 元々呼び出し不可 |
| ex\_timeclear | × | × | 元々呼び出し不可 |
| ex\_timeread2 | × | × | 元々呼び出し不可 |
| ex\_timeevent2 | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_timeread | × | × | 元々呼び出し不可 |
| ex\_rh\_timeevent | ○ | × | ターゲットアクセスできないため不可 |
| ex\_rh\_performancesw | × | × | 元々呼び出し不可 |
| ex\_rh\_performanceclear | × | × | 元々呼び出し不可 |
| ex\_rh\_performanceread | × | × | 元々呼び出し不可 |
| ex\_rh\_performanceset | ○ | × | ターゲットアクセスできないため不可 |

○：呼び出し可、×：呼び出し不可

※ex\_socunitinfo2はEX\_UNIT\_TYPEごとに呼び出し可否が異なる。詳細はex\_socunitinfo2を参照。

[デバッガ関係者向け注意事項]

1. ステータス取得(ex\_getstatus)は使用可能であるが、ユーザプログラムの実行状態のみ状態を反映するが、その他のステータスについては、ターゲットアクセスできないため固定値を返す。詳細はex\_getstatus を参照。
2. 拡張機能使用時は、ハードウェアブレークやソフトウェアブレークなどの他の機能と併用可能。
3. 拡張機能によるブレークは、ex\_runのモードによらず動作する。
4. ex\_stepでステップ実行する際、拡張機能は動作しない。
5. ハードウェアブレークポイント(実行前)、ソフトウェアブレークポイントから再実行する場合、EXEC内部処理としてステップ実行しているが、この処理でも拡張機能は動作しない。

## 外部ソフトトレース単線出力機能に関する動作仕様Operation specifications for external soft trace single-wire output function

[機能概要] [Functional overview]

LPD端子によるソフトトレースメッセージの出力機能の概要について示す。本仕様書内では、本機能を「外部ソフトトレース単線出力機能」と称する。

本機能を使用するには、ex\_socunitinfo2 unit\_type=EX\_UNIT\_TRCSELで有効に設定する必要がある。

本機能を有効にした場合、実行開始後にLPD通信の通常モード(以下「LPD通常モード」)からソフトトレース専用モード(以下「LPDトレースモード」)に切り替え、トレースメッセージをエミュレータへ出力する。ブレーク後にLPD通常モードに切り替わる。LPD通常モード／LPDトレースモードの詳細は下記の通り。

This section gives an overview of the soft trace message output function using the LPD pin. In this specification, this function is called “external soft trace single line output function”.  
  
To use this function, ex\_socunitinfo2 unit\_type = EX\_UNIT\_TRCSEL must be enabled.  
  
When this function is enabled, the LPD communication normal mode (hereinafter “LPD normal mode”) is switched to the soft trace dedicated mode (hereinafter “LPD trace mode”) after execution starts, and the trace message is output to the emulator. Switch to LPD normal mode after a break. Details of LPD normal mode / LPD trace mode are as follows.

・LPD通常モード／LPDトレースモード

LPD通信について、本機能では下記のモードで動作する。

(1)LPD通常モード ：デバイスのDCUやその他資源に対してアクセス可能なモード。本機能が有効の時は、ブレーク中のみこのモードで動作する。なお、本機能が無効の時は、常にこのモードで動作している。

(2)LPDトレースモード ：ソフトトレースのメッセージをエミュレータに出力するモード。このモードの時、デバイス上のDCUやその他資源に対してアクセスすることができないため、EXECの動作が大幅に制限される。本機能が有効なときは、実行中にこのモードで動作する。なお、本機能が無効の時はこのモードで動作することは無い。

・ LPD normal mode / LPD trace mode  
For LPD communication, this function operates in the following modes.  
(1) LPD normal mode: A mode that allows access to the DCU and other resources of the device. When this function is enabled, it operates in this mode only during a break. When this function is disabled, it always operates in this mode.  
(2) LPD trace mode: A mode in which soft trace messages are output to the emulator. In this mode, the DCU and other resources on the device cannot be accessed, so the EXEC operation is greatly limited. When this function is enabled, it operates in this mode during execution. Note that when this function is disabled, it will not operate in this mode.

LPDトレースモード中は強制ブレーク命令とDCUアクセスを行わないEXEC I/F以外は受け付けない。このため、本機能が有効時には、実行中のメモリリード／ライトやトレース開始／停止ができないなど、各機能で制限が発生する。

また、LPDトレースモード中は、デバイスの状態変化を読み出すことができないため、デバイスの状態変化によるコールバックは実行しない。ただし、実行→ブレークの状態遷移は検出可能なため、この時はコールバックが実行される。コールバックを使用せずに実行→ブレークの状態遷移を検出するには、ex\_getstatusを使用する。実行中(LPDトレースモード中)の時は、ex\_getstatus - exestat - es\_usrpro\_run(b0)=1(RUN)となり、他のビットは不定となる。外部ソフトトレース単線出力機能が有効で実行しているときのステータスの詳細はex\_getstatusを参照。

During LPD trace mode, only the forced break instruction and EXEC I / F that does not perform DCU access are accepted. For this reason, when this function is enabled, there are limitations on each function, such as memory read / write being executed and trace start / stop cannot be performed.  
In addition, during LPD trace mode, the device status change cannot be read out, so the callback due to the device status change is not executed. However, since a state transition from execution to break can be detected, a callback is executed at this time. Use ex\_getstatus to detect a state transition from execution to break without using a callback. When executing (in LPD trace mode), ex\_getstatus-exestat-es\_usrpro\_run (b0) = 1 (RUN), and other bits are undefined. Refer to ex\_getstatusex\_getstatusex\_getstatus for details of the status when the external soft trace single line output function is enabled and executing.

本機能で出力されるトレースメッセージは、ユーザプログラムに組み込まれたソフトトレース命令(DBCP命令、DBTAG命令、DBPUSH命令)の実行時に生成されるメッセージだけである。通常のトレース機能とは異なり、分岐命令やデータアクセスの実行時には出力されない。また、スレーブトレース機能は存在しない。

ユーザプログラムに組み込まれたソフトトレース命令の実行頻度によってはエミュレータへのメッセージの格納が間に合わずにロストする場合がある。その場合は、オーバーフローメッセージが保存される。

The trace message output by this function is only a message generated when executing a soft trace instruction (DBCP instruction, DBTAG instruction, DBPUSH instruction) incorporated in the user program. Unlike the normal trace function, it is not output when a branch instruction or data access is executed. There is no slave trace function.  
Depending on the frequency of execution of the soft trace instruction incorporated in the user program, the message may not be stored in time for the emulator to be lost. In that case, an overflow message is saved.

トレースメモリの取得モード(リングモード、フルストップモードなど)、取得したトレースメッセージの読み出し、トレースメモリの消去については、E2拡張機能用のI/Fにより行う。

ただし、E2エミュレータ使用時でも、E2拡張機能が無効に設定されている場合がある。この場合は、下記のEXEC I/Fによって取得モードの設定／参照やトレースメッセージの読み出しを行う。なお、下記のEXEC I/Fは、E2拡張機能が無効の場合に限定して利用できる。

The trace memory acquisition mode (ring mode, full stop mode, etc.), reading of the acquired trace message, and erasing the trace memory are performed by the I / F for the E2 extended function.  
However, even when using the E2 emulator, the E2 extended function may be disabled. In this case, the acquisition mode is set / referenced and the trace message is read using the following EXEC I / F. The following EXEC I / F can be used only when the E2 extended function is disabled.

ex\_rh\_sfttrcrun／ex\_rh\_sfttrcrawmemmode／ex\_rh\_sfttrcrawmeminfo／ex\_rh\_sfttrcrawmemread

上記のEXEC I/F(ex\_rh\_sfttrcrun以外)はブレーク中のみ呼び出し可能で、読み出されるトレースメッセージはブレーク直前までに記録されていたトレースメッセージとなる。再実行時に新たにメッセージの記録を開始するため、その時点までのトレースメッセージは消去される。内蔵トレース機能／外部トレース機能のようにex\_socunitinfo2(EX\_UNIT\_RUNCONTOPT)の設定によって記録を継続するオプションは用意しない。また、ex\_trcclearのように強制的にトレースメッセージを消去するI/Fも用意しない。

The above EXEC I / F (other than ex\_rh\_sfttrcrun) can be called only during a break, and the trace message that is read is the trace message that was recorded immediately before the break. Since new message recording is started at the time of re-execution, the trace message up to that point is deleted. There is no option to continue recording by setting ex\_socunitinfo2 (EX\_UNIT\_RUNCONTOPT) like the internal trace function / external trace function. Also, there is no I / F for forcibly clearing trace messages like ex\_trcclear.

E2拡張機能が有効／無効時に呼び出し可能／不可の外部ソフトトレース単線出力用I/Fについては、「表 2‑9 外部ソフトトレース単線出力用EXEC I/Fの呼び出し可不可モード一覧」を参照。

For the external soft trace single-wire output I / F that can be called / disabled when the E2 extended function is enabled / disabled, see "Table 2 9 Modes for calling external I / F for external soft trace single-wire output Table 2 9 External software Refer to Table 2 9 Execution disable / disable modes of EXEC I / F for trace single line output.

[主な制限事項]

[Main restrictions]

(1)ソフトトレース命令に実行前ブレークのブレークポイントを設定してはならない。設定した場合は、そのブレークポイントから実行した時に、そのソフトトレース命令実行によるソフトトレースメッセージが出力されない。(設定してもエラー応答はしない。)

(1) Do not set a breakpoint for a pre-execution break in a soft trace instruction. If set, the soft trace message by executing the soft trace instruction is not output when the program is executed from the breakpoint. (No error response even if set.)

(2)ソフトトレース命令をステップ実行してはならない。ソフトトレース命令をステップ実行した場合は、そのソフトトレース命令実行によるソフトトレースメッセージが出力されない。 (ステップ実行を行ってもエラー応答はしない。)

(2) Do not step the soft trace instruction. When a soft trace instruction is stepped, no soft trace message is output due to the execution of the soft trace instruction. (No error response even if step execution is performed.)

(3)下記に示すブレークを行った場合、ソフトトレースメッセージ「DBCP(モニタアドレス)」をトレースの最終サイクルに記録する。

(3) When the break shown below is performed, the soft trace message “DBCP (monitor address)” is recorded in the last cycle of the trace.

[該当ブレーク機能]

ソフトウェアブレーク

イベントブレーク

TMU\_TCKタイマ閾値違反ブレーク

パフォーマンスカウント閾値違反ブレーク

[Applicable break function]  
Software break  
Event break  
TMU\_TCK timer threshold violation break  
Performance count threshold violation break

・「DBCP(モニタアドレス)」のトレースメッセージは、ブレークを検出するためにエミュレータ側の制御で生成している。ユーザプログラムからブレーク状態へ遷移した後のモニタ領域でDBCP命令を実行することにより生成する。

-The trace message “DBCP (monitor address)” is generated under the control of the emulator to detect a break. Generated by executing the DBCP instruction in the monitor area after the transition from the user program to the break state.

・「DBCP(モニタアドレス)」のトレースメッセージは、ex\_rh\_sfttrcfilter、ex\_rh\_sfttrcmodeの設定と

関係なく、該当するブレークが発生する場合は常にトレースメッセージを出力する。

DBCP(モニタアドレス)」のトレースメッセージは、PCフィルタリングを使用して常に出力するように

EXEC側で制御しているため、PCフィルタリングの機能をユーザへは公開しない。

-The trace message "DBCP (monitor address)" always outputs a trace message when the corresponding break occurs regardless of the settings of ex\_rh\_sfttrcfilter and ex\_rh\_sfttrcmode.  
      The PCCP function is not disclosed to the user because the control is executed on the EXEC side so that the trace message of “DBCP (monitor address)” is always output using PC filtering.

(4) 外部ソフトトレース単線出力機能のデバイス起因で発生する問題と対策内容を以下に示す。(4) Problems caused by the device of the external soft trace single line output function and the contents of countermeasures are shown below.

表 2‑8 外部ソフトトレース単線出力機能(デバイス起因の問題と対策) Table 2 8 External soft trace single line output function (device-related problems and countermeasures)

|  |  |  |
| --- | --- | --- |
| NO | 問題点problem | EXECでの対応Action in EXEC |
| 1 | トレースメッセージ出力とブレーク要求が同じタイミングで発生した場合、それ以降、ソフトトレースの記録を行うことができなくなる。  If the trace message output and the break request occur at the same timing, the soft trace cannot be recorded thereafter.  [該当条件] [Applicable conditions]  ①～③の条件がすべて成立時。When all the conditions of ① to ③ are met.   1. LPD 1pin 2. 外部ソフトトレース単線出力機能を使用② Use external soft trace single line output function 3. いずれかのブレーク機能が発生One of the break functions occurs   　es\_expfuncact\_brk 拡張機能のアクションブレークExtension action break  es\_expfuncfull\_bk 拡張機能のフルブレークFull break of extensions  es\_sfttrcfull\_bk 外部ソフトトレース単線出力フルブレークExternal soft trace single line output full break  es\_enf\_bk　　　　 強制ブレークForced break | 該当条件が成立後は、下記エラーを通知する  After the conditions are met, the following error is notified。  [対象EXEC I/F] [Target EXEC I / F]  ex\_rh\_sfttrcrawmeminfo  ex\_rh\_sfttrcrawmemread  [エラー] [error]  外部ソフトトレース単線出力機能は記録できない状態を継続中(EX\_DE\_SFTCANTREC)  The external soft trace single line output function is in a state where it cannot be recorded (EX\_DE\_SFTCANTREC) |
| 2 | DeepStopモード中に以下のブレークを行うことができない。  The following breaks cannot be performed while in DeepStop mode.  　 拡張機能のアクションブレーク Extension action break  拡張機能のフルブレークFull break of extensions  外部ソフトトレース単線出力フルブレークExternal soft trace single line output full break  強制ブレークForced break  [該当条件] [Applicable conditions]   1. ～③の条件がすべて成立時。When all the conditions of ① to ③ are met. 2. LPD 1pin 3. 外部ソフトトレース単線出力機能を使用Use external soft trace single line output function 4. DeepStopモードDeepStop mode state | 特になし。nothing special. |
| 3 | 外部ソフトトレース単線出力の有効コア設定は、  デバッガ起動時に1回のみの設定。  The effective core setting for external soft trace single-wire output is set only once when the debugger is started. | 該当条件が成立後は、下記エラーを通知する。After the corresponding condition is satisfied, the following error is notified.  [対象EXEC I/F]  ex\_rh\_sfttrcsw  [エラー]  外部ソフトトレース単線出力機能の出力対象コア指定が既に設定済みのため再設定できません。  (EX\_DE\_ALREADYSFTCORESET)  [error] Cannot be reset because the output target core specification of the external soft trace single line output function has already been set. (EX\_DE\_ALREADYSFTCORESET) |

[ホットプラグイン時の動作]

ホットプラグインRUN中(接続～強制ブレークするまで)は、本機能を有効に設定することができないので、LPDトレースモードで動作することは無い。

[Hot plug-in operation]  
This function cannot be set valid during hot plug-in RUN (from connection to forced break), so it will not operate in LPD trace mode.

[サポート機能]

デバイスによっては対応／非対応の各種機能が存在する。対応／非対応の情報は下記の通りで、ex\_rh\_getenvで取得できる。

・本機能そのもの

・NMで実行されるDBCP命令、DBTAG命令のメッセージ出力の無効設定

・ノンリアルタイムトレースモード

・イベント検出による自動開始／停止 (本開発バージョン未対応)

・TAGフィルタリング機能のグループフィルタリング (本開発バージョン未対応)

[Support function]  
There are various functions that are supported / not supported depending on the device. Correspondence / non-correspondence information is as follows and can be obtained by ex\_rh\_getenvex\_rh\_getenvex\_rh\_getenv.  
・ This function itself  
-Disable message output for DBCP and DBTAG instructions executed by NM  
・ Non-real-time trace mode  
-Automatic start / stop by event detection (This development version is not supported)  
・ Group filtering of TAG filtering function (This development version is not supported)

[対応エミュレータ]

本機能に対応するエミュレータは下表の通り。

なお、本機能の対応／非対応の情報はex\_rh\_getenvで取得できる。

[Supported emulator]  
The following table shows the emulators that support this function.  
Note that information on whether or not this function is supported can be obtained with ex\_rh\_getenvex\_rh\_getenvex\_rh\_getenv.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| エミュレータemulator | E1 | E20 | E2 | IE850 | IE850A |
| 対応／非対応Support/Not support | 非対応  Not support | 非対応  Not support | 対応  support | 非対応  Not support | 非対応  Not support |

[対応コア世代]

本機能に対応するコア世代は下表の通り。

[Supported core generations]  
The following table shows the core generations that support this function.

|  |  |  |
| --- | --- | --- |
| コア世代Core generation | G3 | G4 |
| 対応／非対応  Support/Not support | 対応  Support | 対応  Support |

[マルチコア時の注意事項]

(1)本機能は、マルチコアデバイスであっても複数コアからの同時出力には対応しない。

[Notes on multi-core]  
(1) This function does not support simultaneous output from multiple cores even for multi-core devices.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode.

表 2‑9 外部ソフトトレース単線出力用EXEC I/Fの呼び出し可不可モード一覧

Executable / non-callable modes of EXEC I / F for external soft trace single line output

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EXEC I/F** | **E2拡張機能**  **E2** Extension | **有効**  Effectiveness | | **無効**  **Invalid** | | **備考** |
| **デバッグモード**  Debug mode | **同期**  Sync | **非同期**  asynchronous | **同期**  Sync | **非同期**  asynchronous |
| ex\_rh\_sfttrcsw | | ○ | × | ○ | × |  |
| ex\_rh\_sfttrcmode | | ○ | × | ○ | × |  |
| ex\_rh\_sfttrcevent | | ○ | × | ○ | × | ただし、本開発バージョン未対応  However, this development version is not supported |
| ex\_rh\_sfttrcfilter | | ○ | × | ○ | × |  |
| ex\_rh\_sfttrctune | | ○ | × | ○ | × |  |
| ex\_rh\_sfttrcrun | | × | × | ○ | × | ただし、本開発バージョン未対応  However, this development version is not supported |
| ex\_rh\_sfttrcrawmemmode | | × | × | ○ | × |  |
| ex\_rh\_sfttrcrawmeminfo | | × | × | ○ | × |  |
| ex\_rh\_sfttrcrawmemread | | × | × | ○ | × |  |

○：呼び出し可能、×：呼び出し不可(エラー応答)

○: Can be called, ×: Cannot be called (error response)

## CANブートデバッグに関する動作仕様

[機能概要]

RH850G4世代のE2xシリーズでは、Flash ProgrammerがCAN通信でフラッシュ書き込みをすることが可能である。CAN通信によるフラッシュ書き込みに対応したブートファームをデバッグするための機能を本仕様書内では「CANブートデバッグ」と称する。CANブートデバッグはRH850G4世代のみサポートする。

CANブートデバッグのみシリアルプログラミングモードのデバッグをサポートする。通常動作モードとシリアルプログラミングモード(CANブートデバッグ時)の主な違いを以下に示す。

表 2‑10　通常動作モードとシリアルプログラミングモード時の違い

|  |  |  |  |
| --- | --- | --- | --- |
| 項目 | 通常動作モード | シリアルプログラミングモード | EXECへの影響 |
| FlashのID認証 | CustmerID認証、  DataFlashID認証によるプロテクト可 | プロテクトなし | 解除動作が動いても問題なし |
| PE0以外のメインコア | 動作可能 | 動作不可 | PE0以外を無効に見せる |
| ICUM | 動作可能 | 動作不可 | I付DFとの組み合わせ禁止 |
| ICUM FACI | 使用可能 | 使用不可 | I付DFとの組み合わせ禁止 |
| F-Bist | 実行可能 | 実行不可 | 影響なし |
| RAM 0-ing | OPBT設定に依存 | スキップ | 影響なし |
| RAM保持 | 保持する | 保持しない | 影響なし |

シリアルプログラミングモードへのモードエントリはFlash Programmerが行うため、EXECを含めたデバッガはモードエントリをケアしない。モードエントリされている前提で接続を行う。シリアルプログラミングモードでモードエントリした後のターゲットマイコンに対しホットプラグイン接続を行う。CANブートデバッグ時は、必ずホットプラグインモードで接続すること。また、ex\_socunitinfo2コマンドのunit\_type = EX\_UNIT\_LPDOPTでCANブートデバッグを指定する。

CANブートデバッグではシリアルプログラミングモードで動作中のターゲットに接続するため、以下の点が通常動作モード時と異なる。

1. PE0以外のコアは動作しない
2. リセット禁止
3. Flash資源の書き換え禁止

シリアルプログラミングモードのデバッグであるため、PE0以外のコアは動作しない。PE0以外のコアを選択しようとした場合、EX\_PE\_INVALIDCORESELECTエラーを返す。マルチコアであってもシングルコアとして扱う。ICUMも動作しないため、ICUMありのDFファイルとの組み合わせもサポートしない。

リセットが発生するとモードエントリから再開する必要があるため、リセット入力は禁止とする。CANブート接続時はex\_reset、ex\_resetrunコマンドを呼ぶとEX\_SE\_NOSPRT\_DURING\_CANBOOTエラーを返す。また、ex\_exitコマンドによる終了時は、パラメータに関係なくリセット解除状態で終了する。

ブートファームがFlash書き換えを行うため、デバッガからFlash資源に対する書き込みは禁止する。書き換えようとした場合、EX\_SE\_NOSPRT\_DURING\_CANBOOTエラーを返す。ソフトウェアブレークはケアしない。ブートファームはRAM実行するため、Flash資源に対するソフトウェアブレークが設定されない想定である。RAMに対するソフトウェアブレークは設定可能。

## 初期停止・スタンバイモードデバッグ環境に関する動作仕様

[機能概要]

初期停止状態・Cyclicモード(CyclicRun・CyclicStop)期間中のデバッグ機能をサポートする。

サポートするためにはデバイス内部の特定デバッグ機能が必要なため、RH850 G4コア世代デバイスは全て、G3コア世代では一部のデバイスのみサポートする。

サポートが可能なデバイスは、デバイスファイルのEFIタグでfunc:0x0103のbit3(Initial Stop Debug)を確認することにより判定できる。

EFI部の詳細についてはデバイスファイルの仕様書を確認すること。

本デバッグ環境および従来のデバッグ環境の両方をサポートするため、エミュレータ接続手順の中で以下のI/Fを発行することによってデバッグ環境の切り替えを行う。なお、デバッグ環境の切り替えは、接続完了後から切断までの期間中は行えない。

　ex\_socunitinfo2　unit\_type = EX\_UNIT\_INITSTOPDBG

初期停止・スタンバイモードデバッグ環境を有効にして接続した場合、従来のデバッグ環境と比較して以下に示す違いがある。

・エミュレータ接続手順完了直後から初期停止コアを維持した状態でデバッグを開始することができる。

・リセットコマンド(ex\_reset)で指定するパラメータ(mode)の動作が変わる。

mode=EX\_RESET, EX\_FRESET　　　　 ：初期停止となるコアは、処理完了後に初期停止状態を維持

mode=EX\_INITSTOP\_BRKRES(新規追加)：初期停止となるコアも、処理完了後に起動させる(従来のデバッグ環境と同様)

・CyclicRun中のメインコア・Cyclic Disable中のメインコア以外のコアステータス状態は、プログラム実行中だけでなくユーザプログラム停止中も値が有効である。ただし、ICUM有効の場合、CyclicRun中のメインコア・Cyclic Disable中のメインコア以外のコアステータス状態は、プログラム実行中だけ存在する。

・初期停止中のコアに対するステータス状態は、ユーザプログラム実行中だけでなくユーザプログラム停止中も存在する。また、プログラム停止中にステータス状態が、初期停止中から起動中、起動中から初期停止中に変化する。

・初期停止中のコア・CyclicDisable中のメインコア以外のコアを選択している場合、デバイス内部レジスタに対するアクセスができないため、EXEC I/Fの一部に制限がかかる。

・Stop、DeepStop、メインコアがCyclicStop、メインコア以外のコアがCyclicDisable状態の場合、強制リセットを除くほとんどのデバッグ機能に関するEXEC I/Fが使用できない。

・非同期デバッグモードへの切り替えができない。

・外部ソフトトレース単線出力機能、E2拡張機能は使用できない。

表 2‑11～表 2‑12に制御対象のCPUステータスとEXEC I/F発行の制限について記載する。

[デバッガ関係者向け注意事項]

初期停止中のコアは、デバイスのBOOTCTRLレジスタに有効にする値をライトすることにより、初期停止が解除される。このため、ユーザプログラム停止中に、初期停止中のコアが存在しても、BOOTCTRLレジスタに有効にする値をライトするEXEC I/Fコマンドを発行することにより、初期停止状態が解除される可能性がある。

そのためデバッガは、ユーザプログラム停止中であってもCPUステータスを参照するEXEC I/Fコマンド(ex\_getstatus)を定期的に発行してCPUステータスの状態変化をチェックすることを推奨する。(EXECの非同期通信インターフェースを有効にしている場合は、CPUステータスに変化が発生すると、EXECがコールバック関数を使用してデバッガに通知する)

表 2‑11 デバッグ環境のモードの違いよるEXEC I/F発行可否(同期デバッグモード、ブレーク中)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 機能 | I/F関数 | ex\_socunitinfo2(EX\_UNIT\_COREPESEL)選択コアのCPUステータス | | | | | |
| 従来のデバッグ環境 | 初期停止・スタンバイモードデバッグ環境が有効 | | | | |
| 状態区別なし | 通常 | 初期停止 | メインコアがCyclic Run  ※10 | メインコア以外のコアがCyclic Disable  ※10 | Stop,  DeepStop,  メインコアがCyclicStop,  メインコア以外のコアがCyclicDisable |
| 実行 | ex\_run | ○ | ○ ※7 | ×　※1 | ○ ※7,8 | ×　※1 | － |
| ブレーク | ex\_stop | － | － | － | － | － | － |
| ステップ | ex\_step | ○ | ○　※9 | ×　※1 | ○ | ×　※1 | － |
| リセット | ex\_reset | ○ | ○ | ○ | ○ | ○ | － |
| リセットGO | ex\_resetrun | ○ | ○ | ○ | ○ | ○ | － |
| メモリ | ex\_memwrite  (Flash資源) | ○ | ○　※2 | ○　※2 | ×　※3 | ×　※3 | － |
| ex\_memwrite  (Flash資源以外) | ○ | ○ | ○　※4 | ○ | ○　※4 | － |
| ex\_memread  (Flash資源) | ○ | ○ | ○ | ○ | ○ | － |
| ex\_memread  (Flash資源以外) | ○ | ○ | ○　※4 | ○ | ○　※4 | － |
| ex\_memfill | ○ | ○ | ○　※4 | ○ | ○　※4 | － |
| ex\_memcopy | ○ | ○ | ○　※4 | ○ | ○　※4 | － |
| レジスタ | ex\_regwrite | ○ | ○ | ×　※1 | ○ | ×　※1 | － |
| ex\_regread | ○ | ○ | ×　※1 | ○ | ×　※1 | － |
| イベント・  ブレーク | ex\_rh\_evncond  (コアイベント) | ○ | ○ | ×　※1 | ○ | ×　※1 | － |
| ex\_hwbrkevent  (コアイベント) | ○ | ○ | ×　※1 | ○ | ×　※1 | － |
| SWブレーク | ex\_swbrkcond  (Flash資源) | ○ | ○ | ○ | ×　※5 | ×　※5 | － |
| ex\_swbrkcond  (RAM資源) | ○ | ○ | ○ | ○ | ○ | － |
| ハードウェア設定 | ex\_socunitinfo2  (同期・非同期デバッグモードの切り替え) | ○ | ×　※6 | ×　※6 | ×　※6 | ×　※6 | － |
| トレース | トレースの全I/F | ○ | ○ | ○ | ○ | ○ | － |
| タイマ・パフォーマンス | タイマ・パフォーマンスの全I/F | ○ | ○ | ○ | ○ | ○ | － |
| 外部ソフトトレース単線出力 | 外部ソフトトレース単線出力の全I/F | × ※6 | × ※6 | × ※6 | × ※6 | × ※6 | － |
| E2拡張 | E2拡張の全I/F | × ※6 | × ※6 | × ※6 | × ※6 | × ※6 | － |

○：使用可能　×：使用不可　－：この動作モードでは未サポート

※1：CPUが停止(初期停止・Cyclic Disable)中のためコマンド実行できない旨のエラーを返す。

なお、ex\_regreadコマンドでPC、SP(R3)、FP(R28)、LP(R31)の各レジスタを単独で読み出した場合は、ダミー値(0x00000000)を返してエラーとしない。

※2：G3世代ではCPU1(G4世代ではCPU0)含めて全てのコアが初期停止中の場合は、Flash資源への書き込みができない旨のエラーを返す。

※3：Cyclicモード中のためFlash資源への書き込みコマンドの実行ができない旨のエラーを返す。

※4：アクセス領域がMAUアクセス可能な領域の場合は、コマンドを実行して正常終了を返す。ただし、CPUアクセスが必要な領域の場合は、CPUが停止(初期停止・Cyclic Disable)中のため指定された領域へアクセスできない旨のエラーを返す。

※5：Cyclicモード中のためコマンド実行できない旨のエラーを返す。

※6：本デバッグ環境モードでは、機能自体が使用できない。関連するI/Fコマンドは、サポートしていない旨のエラーを返す。

※7：コア(実行前)イベントについては、停止(初期停止・Cyclic Disable)中のコアに対して設定・変更ができないため、動作しているコアに対してのみ設定・変更を行なって実行を開始する。

※8：Flash資源に対するソフトウェアブレークについて、Cyclicモード中はFlash資源への書き込みができないデバイス仕様によってソフトウェブレークの設定・変更ができないため、Flash資源に対するソフトウェアブレークの設定・変更を行わないで実行を開始する。

※9：停止(初期停止・Cyclic Disable)中のコアに対してはステップ実行を行わない。ただし、初期停止中でないコアの連続ステップ実行により、初期停止中のコアが解除された場合は、初期停止中だったコアについても初期停止解除後からステップ実行を行なう。

※10：ICUM有効の場合、ブレーク中の状態に遷移しないため未サポート

表 2‑12デバッグ環境のモードの違いよるEXEC I/F発行可否(同期デバッグモード、実行中)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 機能 | I/F関数 | ex\_socunitinfo2(EX\_UNIT\_COREPESEL)選択コアのCPUステータス | | | | | |
| 従来の  デバッグ環境 | 初期停止・スタンバイモードデバッグ環境が有効 | | | | |
| 状態区別なし | 通常/Halt | 初期停止 | メインコアがCyclic Run  ※6 | メインコア以外のコアがCyclic Disable  ※6 | Stop,  DeepStop,  メインコアがCyclicStop,  メインコア以外のコアがCyclicDisable |
| 実行 | ex\_run | － | － | － | － | － | × ※4 |
| ブレーク | ex\_stop | ○ | ○ | ○ | ○ | ○ | × ※4 |
| ステップ | ex\_step | － | － | － | － | － | × ※4 |
| リセット | ex\_reset | ○ | ○ | ○ | ○ ※7 | ○ ※7 | ○ ※7 |
| リセットGO | ex\_resetrun | － | － | － | － | － | × ※4 |
| メモリ | ex\_memwrite  (Flash資源) | － | － | － | － | － | × ※4 |
| ex\_memwrite  (Flash資源以外) | ○ | ○ | ○　※1 | ○ | ○　※1 | × ※4 |
| ex\_memread  (Flash資源) | ○ | ○　※3 | ○　※3 | ○　※3 | ○　※3 | × ※4 |
| ex\_memread  (Flash資源以外) | ○ | ○ | ○　※1 | ○ | ○　※1 | × ※4 |
| ex\_memfill | ○ | ○ | ○　※1 | ○ | ○　※1 | × ※4 |
| ex\_memcopy | － | － | － | － | － | × ※4 |
| レジスタ | ex\_regwrite | ○ | ○ | ×　※2 | ○ | ×　※2 | × ※4 |
| ex\_regread | ○ | ○ | ×　※2 | ○ | ×　※2 | × ※4 |
| イベント・  ブレーク | ex\_rh\_evncond  (コアイベント) | ○ | ○ | ×　※2 | ○ | ×　※2 | × ※4 |
| ex\_hwbrkevent  (コアイベント) | ○ | ○ | ×　※2 | ○ | ×　※2 | × ※4 |
| SWブレーク | ex\_swbrkcond  (Flash資源) | － | － | － | － | － | － |
| ex\_swbrkcond  (RAM資源) | － | － | － | － | － | － |
| ハードウェア設定 | ex\_socunitinfo2  (同期・非同期デバッグモードの切り替え) | － | － | － | － | － | － |
| トレース | トレースの実行中に発行可能なI/F | ○ | ○ | ○ | ○ | ○ | × ※4 |
| タイマ・パフォーマンス | タイマ・パフォーマンスの実行中に発行可能なI/F | ○ | ○ | ○ | ○ | ○ | × ※4 |
| 外部ソフトトレース単線出力 | 外部ソフトトレース単線出力の全I/F | ○ | × ※5 | × ※5 | × ※5 | × ※5 | × ※5 |
| E2拡張 | E2拡張の全I/F | ○ | × ※5 | × ※5 | × ※5 | × ※5 | × ※5 |

○：使用可能　×：使用不可　－：この動作モードでは未サポート

※1：MAUアクセス可能な領域を指定した場合は可能。一時ブレークによるアクセスを指定した場合は、CPUが停止(初期停止・Cyclic Disable)中のためコマンド実行できない旨のエラーを返す。

※2：CPUが停止(初期停止・Cyclic Disable)中のためコマンド実行できない旨のエラーを返す。

※3：一時ブレークによるアクセスを指定した場合のみ可能。

※4：デバッグ操作を行なえないスタンバイモード中のためコマンド実行できない(強制リセットを促す)旨のエラーを返す。

※5：本デバッグ環境モードでは、機能自体が使用できない。機能を有効にするためのI/Fコマンドではサポートしていない旨のエラーを返し、機能有効後に使用するI/Fコマンドは、機能が有効でない旨を示す従来のエラーを返す。

※6：ICUM有効の場合、デバッグ操作を行なえないスタンバイモード中のためコマンド実行できない(強制リセットを促す)旨のエラーを返す。

※7：ホットプラグイン接続している場合、ホットプラグイン接続後の該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行できない(スタンバイモードが解除された状態での操作を促す)旨のエラーを返す。

[主な制限事項]

初期停止・スタンバイモードデバッグ環境が有効時は、デバイス内デバッグモジュールの仕様によって回避できない動作があり、以下に示す制限が発生する。

ユーザに知らせる必要がある制限については、マニュアルに記載してガイドする。

【実行・ステップ実行】

・初期停止中のコア・Cyclic Disable中のメインコア以外のコアを選択している場合、実行・ステップ実行コマンドは受け付けずエラー応答する。

【システムレジスタアクセス】

・初期停止中のコア・Cyclic Disable中のメインコア以外のコアを選択している場合、汎用・システムレジスタへのリード/ライトコマンドは受け付けずエラー応答する。ただし、汎用・システムレジスタへのリードコマンドでPC、SP(R3)、FP(R28)、LP(R31)の各レジスタを単独で指定した場合は、ダミー値(0x00000000)を返してエラーとしない。

【メモリアクセス】

・初期停止中のコア・Cyclic Disable中のメインコア以外のコアを選択して、CPUアクセスが必要な領域を指定したリード/ライトコマンドは受け付けずエラー応答する。ただし、MAUアクセスが可能な領域を指定した場合は正常処理を行いエラー応答しない。

CPUアクセスを使用する領域は、G3世代デバイスでは、外部空間領域、未定義領域、I/O領域であり、G4世代デバイスでは、I/O領域の一部となる。

なお、RAM領域・I/O領域のSelf領域については、EXEC内部でSelf領域でないRAM領域・I/O領域にアドレスを変換してMAUアクセスしてエラー応答しない。

【フラッシュメモリアクセス】

・CPUステータスが、メインコアがCyclicRun中・メインコア以外のコアがCyclicDisable中の場合、フラッシュメモリへの書き換えは行えないため対応するコマンドは受け付けずエラー応答する。

・G3世代ではCPU1(G4世代ではCPU0)のコアが初期停止中の場合、フラッシュメモリへの書き換えは行えないため対応するコマンドは受け付けずエラー応答する。

【ソフトウェアブレーク】

・CPUステータスが、メインコアがCyclicRun中・メインコア以外のコアがCyclicDisable中の場合、フラッシュメモリ領域へのソフトウェアブレーク設定/解除コマンドは受け付けずエラー応答する。

・ソフトウェアブレークコードがコードフラッシュ領域に埋め込まれた状態で、ユーザプログラム停止中に遷移し、CPUステータスが、メインコアがCyclicRun中・メインコア以外のコアがCyclicDisable中となっていた場合、次回以降の実行でフリー実行を選択してもソフトウェアブレークコードをユーザコードに書き戻しができない。

　逆に、ソフトウェアブレークが設定済みの状態で、一度フリー実行を行い、その後ユーザプログラム停止中に遷移し、CPUステータスが、メインコアがCyclicRun中・メインコア以外のコアがCyclicDisable中となっていた場合、次回以降の実行でブレーク付き実行を選択してもユーザコードをソフトウェアブレークコードに書き換えができない。

　これらの場合は、ソフトウェアブレークコードの埋め込み状態を変更せずに実行を開始するため、意図したソフトウェアブレークが発生しない、意図しないソフトウェアブレークが発生する可能性がある。(ブレーク要因としてはソフトウェアブレーク)

・デバイスへのソフトウェアブレーク設定と実行状態の不整合は、CPUステータスが、メインコアがCyclicRun中・メインコア以外のコアがCyclicDisable中から解除されて通常の状態に戻った次の実行コマンドで正常に戻る。

【CPUコア(実行前)イベントブレーク】

・初期停止中のコア・Cyclic Disable中のメインコア以外のコアを選択している場合、CPUコア(実行前)イベントブレークの設定/削除コマンドは受け付けずエラー応答する。

・CPUコア(実行前)イベントがデバイスに設定された状態で、ユーザプログラム停止中に遷移し、初期停止中・Cyclic Disable中となっていた場合、起動中のコアの次回以降の実行でフリー実行を選択しても、初期停止中のコア・Cyclic Disable中のメインコア以外のコアに対するCPUコア(実行前)イベントを解除できない。

　逆に、CPUコア(実行前)イベント条件が設定済みの状態で、一度フリー実行を行い、その後ユーザプログラム停止中に遷移し、CPUステータスが、初期停止中・Cyclic Disable中となっていた場合、次回以降の実行でブレーク付き実行を選択しても初期停止中のコア・Cyclic Disable中のメインコア以外のコアに対してはデバイスにCPUコア(実行前)イベントを設定できない。

これらの場合は、デバイスへのCPUコア(実行前)イベント設定の状態を変更せずに実行を開始するため、意図したイベントブレークが発生しない、意図しないイベントブレークが発生する可能性がある。

・デバイスへのCPUコア(実行前)イベント設定と実行状態の不整合は、初期停止中・Cyclic Disable中から解除されて通常の状態に戻った次の実行コマンド、もしくはリセットコマンドで正常化する。

【同期/非同期デバッグモードの切り替え】

・初期停止・スタンバイモードデバッグ環境が有効の場合、非同期デバッグモードへの切り替えは行えない。

【ホットプラグイン接続】

・ホットプラグインRUN中に強制ブレークコマンドを発行した場合、デバイス仕様の影響により制限を受ける。

デバイスにG3KコアまたはG3K-Hコアが実装されていて、初期停止中のコア・Cyclic Disable中のメインコア以外のコアがある場合、強制ブレーク後の再実行時に必要なデバッグ機能の制御が行えない。

ユーザのデバッグ操作が理解しやすいことを優先して、G3KコアまたはG3K-Hコアの実装有無にかかわらず、本デバッグ環境をサポートする全てのデバイスで以下の対応に統一する。

・初期停止中のコア・CyclicDisable中のメインコア以外のコアが存在する場合はコマンドを受け付けずエラー応答する。

【メインコアデバッグ】

・G3世代ではCPU1(G4世代ではCPU0)のコアが初期停止中の場合、実行・ステップ実行、フラッシュメモリの書き換え、ソフトウェアブレーク設定・削除、CPUコア(実行前)イベント設定・削除、汎用・システムレジスタの読み出し・書き換え、CPUアクセスを伴う領域へのメモリアクセスなどのコマンド発行はエラー応答する。

【初期停止・CyclicRun以外のスタンバイモード】

CPUステータスが初期停止・CyclicRun以外のスタンバイモード(Stop・DeepStop・CyclicStop・CyclicDisable)だった場合、デバッグモジュールに対するアクセスが行えない。そのため、トレース、タイマ、パフォーマンスなどほとんどのデバッグ機能に関するEXEC I/Fを発行した場合は受け付けずエラー応答する。

# EXECインターフェース関数仕様

本章は、各EXECインターフェース関数仕様の詳細説明を行う。

## 記述規定

インターフェース関数仕様の記述方法に付いて説明する。

記述形式は図 3‑1のようになる。

|  |  |  |
| --- | --- | --- |
|  | **Function Name Status** |  |
|  |  |  |
|  | **[Function]** |  |
|  |  |  |
|  |  |  |
|  | **[Format]** |  |
|  |  |  |
|  |  |  |
|  | **[Arguments]** |  |
|  |  |  |
|  |  |  |
|  | **[Return Values]** |  |
|  |  |  |
|  |  |  |
|  | **[Description]** |  |
|  |  |  |
|  |  |  |

図 3‑1関数説明の構成

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function name** | **:** | 関数名が記載されている。 | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | | | | |
| **Status** | **:** | EXECの状態と関数の呼び出し可/不可の関係を表形式で記載する。 | | | | | | | | | | | | | | |
|  |  | SYNC | | EMU | ○ | BRK | × | TRC | ○ | TIM | ○ | COV | ○ | HOT | ○ |
|  |  | ASYNC | | EMU1 | ○ | EMU2 | × | BRK1 | ○ | BRK2 | × | TRC1 | ／ | TRC2 | － |
|  | ※ただし、関数によっては条件によって変わる場合がある。その場合は、条件ごとに表を追加する。 | | | | | | | | | | | | | | |
|  | ＜EXECの状態(非同期のステータス)＞ | | | | | | | | | | | | | | |
| SYNC：同期デバッグモード時 | | | | | | | | | | | | | | |
|  | EMU： | | ユーザプログラム実行中 | | | | | | | | | | | |
|  | BRK： | | ブレーク中 | | | | | | | | | | | |
|  | TRC： | | トレース起動中 | | | | | | | | | | | |
|  | TIM： | | タイマ動作中(複数あるチャネルのうちいずれかひとつのタイマでも計測していれば計測中となる。) | | | | | | | | | | | |
|  | COV： | | カバレッジ機能動作中 | | | | | | | | | | | |
|  | HOT： | | ホットプラグインRUN中(接続～強制ブレークするまで) | | | | | | | | | | | |
| ASYNC：非同期デバッグモード時 | | | | | | | | | | | | | | |
|  | EMU1： | | 全コアが実行中 | | | | | | | | | | | |
|  | EMU2： | | カレントコアは実行中だが、他コアのうち1個以上がブレーク中 | | | | | | | | | | | |
|  | BRK1： | | 全コアがブレーク中 | | | | | | | | | | | |
|  | BRK2： | | カレントコアはブレーク中だが、他コアのうち1個以上が実行中 | | | | | | | | | | | |
|  | TRC1： | | カレントコアがトレース取得中(カレントコアはデバッグ対象コア) | | | | | | | | | | | |
|  | TRC2： | | 他コアがトレース取得中(カレントコアはデバッグ非対象コア) | | | | | | | | | | | |
|  | ※TIMは呼び出し可/不可に影響を与えないため省略  ※COVは未サポート機能であるため省略  ※HOTは同期デバッグモード時のみサポートのためのため省略 | | | | | | | | | | | | | |
|  |  | |  | | | | | | | | | | | |
|  | ＜呼出し可/不可＞ | | | | | | | | | | | | | |
|  | ○： | 関数呼び出し可。条件によってはエラーとならない場合も含む | | | | | | | | | | | | |
|  | ×： | 関数呼び出し時、必ずエラーとなる | | | | | | | | | | | | |
|  | ／： | Status対象外  例)  ・EMUが"×"の場合、TRCやTIMのステータスは"○"になることはないため"／"となる  ・EMU1が"×"の場合、TRC1は"○"になることはないため"／"となる | | | | | | | | | | | | |
|  | －： | EXECの状態に関わらずEX\_NOSUPRTを返す。 | | | | | | | | | | | | |
|  | |  |  | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function** | **:** | 関数機能の概略が記載されている。 | | | | | | |
|  |  |  | | | | | | |
| **Format** | **:** | 関数の呼び出し形式と引数の入出力属性および簡単な説明を記述する。以下に記述例を示す。  int ex\_function( UCHAR param1, USHORT param2, ULONG \*param3)  型 引数識別名 属性 説明 対象  ------------------------------------------------  UCHAR param1 <I > param1の説明 ○  USHORT param2 <I > param2の説明 --  ULONG \*param3 <IO> param3の説明 ○  型: 引数型を以下のデファインで記述する。  UCHAR: unsigned char  USHORT: unsigned short  ULONG: unsigned long  属性: 入出力属性を以下の様に記述する。  <IO>: I :入力で使用する引数を意味する。  O :出力で使用する引数を意味する。  <-->: --:使用しない場合を意味する。  対象: 使用有無を以下の様に記述する。  ○:使用する  --:使用しない | | | | | | |
|  |  |  | | | | | | |
| **Arguments** | **:** | 各引数の入力範囲を記述。また、直接の値ではなく名称で記述されている場合は3.1.1の入力範囲規定を参照のこと。例を以下に示す。  またポインタの示す領域は、呼び出し側で用意すること。 | | | | | |
| param1 | : | EX\_DEF\_A  EX\_DEF\_B | | : | 入力範囲は、EX\_DEF\_AとEX\_DEF\_Bとなる。範囲外の入力はエラーとなる。 |
| param2 | : | <1 - FFFFh> | | : | 入力範囲は1-FFFFhとなる。 |
| \*param3 | : | <address value> | | : | 3.1.1の入力範囲規定を参照。 |
|  |  |  | | | | | |
| **Return Values** | **:** | 戻り値(define名)とその内容を記述。また、戻り値にはここに記述する内容の他にあらゆる関数(通信処理等の要因)で発生するエラー(戻り値)がありその内容は以下の通りである。 | | | | | |
| EX\_FE\_SENDOUT | | | 別の上位関数からのコマンド実行が完了しなかった。 | | |
| EX\_FE\_MONTIMEOUT | | | モニタの処理が時間内に終了しなかった。 | | |
| EX\_DE\_M\_DCUACCESS | | | DCUのアクセス(JTAG通信)に失敗した | | |
| EX\_DE\_M\_DCUACCVERIFY | | | DCUのアクセス(JTAG通信)に失敗した  (ベリファイエラー) | | |
| EX\_UE\_M\_BUSHOLDCONT | | | バスホールド継続中(モニタタイムアウト系エラー) | | |
| EX\_SE\_TRGTOFF | | | ターゲット電源OFF中 | | |
| EX\_FE\_USBFAIL | | | USB通信が切れた(ソフトを終了させて、ICEの電源を入れなおす必要がある) | | |
| EX\_FE\_ANOTHEREXECRUN | | | ディレクトリが異なる別のEXECが既に動作している(USBドライバ競合の危険がある) | | |
| EX\_FE\_EXECABORT | | | 内部エラー(EXEC内部処理で矛盾が生じた場合) | | |
|  |  | EX\_QC\_INTERNALERROR\_PENOMATCH | | | EXEC内部エラー。指定カレントコア番号と異なるコア番号をEXEC内部管理変数に格納した。 | | |
|  |  |  | | | | | |
| **Description** | **:** | 関数の機能説明と制限等を記述。 | | | | | |
| 改版の項目削除の際は、前版からの削除を明確にするため二重取り消し線で取り消し後、次版で消す。 | | | | | | | |

### 名称で示される入力範囲規定

|  |  |  |  |
| --- | --- | --- | --- |
| Address value | : | 00000000h - FFFFFFFFh | |
|  |  |  | |
| Event number | : | 0001h - FFFFh(0000hを除く) | |
|  |  | イベント番号には、ex\_rh\_evncondで指定した番号(Single event number)とex\_rh\_evnintgcondで指定した番号(Integrated event number)が指定できる。 | |
|  |  |  | |
| Single event number | : | 0001h - 7FFFh |
| CPU event number | : | 0001h - 3FFFh |
| Slave event number |  | 4000h – 7FFFh  (<TEU\_GRM/TEU\_CRM>, <TEU\_GVC/TEU\_KVC/TEU\_AXI/TEU\_DMA>, <TEU\_EMS>, <TEU\_LRM>用のイベント番号) |
|  |  |  |
| Integrated event number | : | 8000h - FFFFh |
|  |  |  |
| Register ID | : | レジスタIDは図 3‑2に示すような意味を持つ。  (レジスタIDの一覧は3.4を参照) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ＜Program register＞ | | | | | |
| 汎用レジスタ(r0 －r31) | |  | | | |
|  | 汎用レジスタ番号 |  | r0 - r31を0x0000 - 0x001Fに割り当てる | | |
| プログラムカウンタ(PC) | |  |  | | |
|  | 専用番号 |  | PC専用番号として0x0020を割り当てる | | |
| ベクトルレジスタ(vr) | |  |  | | |
|  | ベクトルレジスタ番号 |  | vr0 - vr31を0x0030 - 0x004Fに割り当てる | | |
| 0x0000 - 0x00FFは、Program register番号として使用する。 | | | | | |
|  | | | | | |
| ＜System register＞ | | | | | |
| 0x0100 + (selID << 5) + regID | |  |  |  |  |
|  | |  | selID | : | システムレジスタ番号 |
|  | |  | regID | : | システムレジスタのグループ番号 |
| selID、regIDの範囲: 0x00 - 0x1F | | | | | |

図 3‑2 レジスタIDの構成

|  |  |  |  |
| --- | --- | --- | --- |
| Access size | | : | サポートするアクセスサイズを表 3‑1に示す。 |
|  |  | | |

表 3‑1 アクセスサイズ使用状況一覧

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Access size  Device | 1bit  EX\_DATA\_SIZE\_B | 1byte  EX\_DATA\_SIZE\_1 | 2byte  EX\_DATA\_SIZE\_2 | 4byte  EX\_DATA\_SIZE\_4 | 8byte  EX\_DATA\_SIZE\_8 | All access size  EX\_DATA\_SIZE\_NC |
|  | RH850 | × | ○ | ○ | ○ | ○ | ○ |

## 関数詳細説明

関数別の詳細説明は、RH850で使用できるもののみ載せる。

### 起動、終了、リセット

起動時における初期化、終了、リセットのための関数

ex\_exit, ex\_initexec, ex\_reset, ex\_setemulator, ex\_getemulist, ex\_extpower

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_exit | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

EXECを終了する

[Format]

int ex\_exit( USHORT mode )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 終了モード | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | EX\_END\_PROC | 終了処理を行い、EXECを終了する。  (ターゲットへのRESET入力状態にする) |
|  | EX\_END\_PROC\_WO\_RESET | 終了処理を行い、EXECを終了する。  (ターゲットへのRESET解除状態にする)  E2/IE850Aのみサポートする。 |
|  | EX\_END\_COMM | EXECを終了するのみ。  (ユーザ非公開：EXECとICEのデバッグに使用) |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

[Description]

[引数説明]

modeは、EX\_END\_PROCまたはEX\_END\_PROC\_WO\_RESETを指定すること。ただし、EX\_END\_PROC\_WO\_RESETはE2/IE850Aエミュレータのみでサポート。他のエミュレータで呼び出した場合はEX\_PE\_DATAが返る。

[動作説明]

EXECを終了させる。

EXECの起動にはex\_initexec()を使う。ex\_initexec()が正常終了したら、必ずex\_exit()で終了すること。正常終了しない場合も、電源供給機能が動作し、エミュレータとの接続処理が動くため必ずex\_exit()を呼び出してEXECをデタッチすること。

E1/E20エミュレータの場合は、ターゲット接続ケーブルのリセット端子をアクティブに設定する。ただし、最終的にはターゲット接続ケーブルの結線をどのようにユーザが設計しているかに依存する。また、EXECはBFWをLv0に遷移させ、Communi.dllをアンロードしてUSB切断まで行う。

E2エミュレータ/IE850Aエミュレータの場合はmodeの指定によりリセット端子をアクティブにするか、インアクティブにするかを選択できる。CANブートデバッグ時はパラメータの指定を無視し、必ずインアクティブで終了する。E1/E20エミュレータと同様に、BFWをLv0に遷移させ、Communi.dllをアンロードしてUSB切断まで行う。

FULLエミュレータの場合は、デバッグチップにリセットを入れた状態で終了する。OCDの場合と同様にUSB切断まで行う(BFW、Communi.dllは使用していない)。

ex\_initexecコマンド、またはex\_extpowerコマンドによりターゲット電源供給機能を有効にしている場合は、ex\_exitコマンド実行時に電源の供給を停止する。ただし、IE850Aエミュレータの場合は、電源の供給を継続する。

[デバッガ関係者向け注意事項]

1. IE850接続時にex\_exitコマンドを実行したとき、リセットベクタアドレスに対して自番地にジャンプする命令が書き込まれる。ただし、D1M2 1stCutのPODのみ対象とする。
2. デバイスの品種によっては、デバッグ通信I/Fの切り替えをオプションバイトの書き換えではなく、デバッグ通信I/F端子を制御することで行なう。このデバイス品種の場合、起動時に指定した終了時のデバッグ通信I/Fモード指定(ex\_socunitinfo2(EX\_UNIT\_LPDOPT unit\_info[1]))は無視して、オプションバイトの書き換えは行わない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_initexec | **EXEC起動前** | **EMU** | **○** | **BRK** | **○** | **TRC** | ／ | **TIM** | ／ | **COV** | ／ | **HOT** | ○ |
| **EXEC起動後** | **EMU** | **×** | **BRK** | **×** | **TRC** | **×** | **TIM** | **×** | **COV** | ／ | **HOT** | **×** |
| **※本関数は起動前しか呼べないため、同期/非同期は関係しない** | | | | | | | | | | | | |

[Function]

EXECを起動する

[Format]

int ex\_initexec( char \*hostname, USHORT portno, EXINF \*ex\_inf )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| char | \*hostname | <--> |  | -- |
| USHORT | portno | <--> |  | -- |
| EXINF | \*ex\_inf | <I > | 起動情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | char | \*exec\_name | <--> |  | -- |
|  | EXDEVDAT | reg\_inf | <--> |  | -- |
|  | EXUSERDAT | user\_inf | <--> |  | -- |
|  | HWND | hWnd | <--> | NULL | -- |
|  | HANDLE | hAccTable | <--> |  | -- |
|  | char | \*devfile\_inf | <I > | デバイスファイル名(フルパス) | ○ |
| }EXINF | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | \*devfile\_ptr | <--> |  | -- |
|  | USHORT | data\_num | <--> |  | -- |
| }EXDEVDAT | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | vdd\_mode | <I > | ホットプラグイン起動指定、電源供給指定用(OCD) | ○ |
|  | UCHAR | clk\_mode | <--> |  | -- |
|  | UCHAR | flash\_mode | <--> |  | -- |
|  | UCHAR | pin\_mode | <--> |  | -- |
|  | ULONG | mask\_fg | <--> |  | -- |
|  | ULONG | sel\_fg | <--> |  | -- |
| }EXUSERDAT | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| hWnd | | |
|  |  | NULL |
|  |  |  |
| devfile\_inf | | |
|  |  | デバイスファイル名のポインタ |
|  |  |  |
| user\_inf.vdd\_mode | | |
|  | EX\_VDD\_NORMAL | 通常の状態 |
|  | EX\_VDD\_TARGET\_HA | ホットプラグイン起動(OCD)  IE850、IE850Aエミュレータではサポートしない |
|  | EX\_VDD\_5V\_OUT | E1、E2エミュレータをVDD=5V出力で動作させる場合(OCD)  E20、IE850、IE850Aエミュレータではサポートしない |
|  | EX\_VDD\_3V\_OUT | E1、E2エミュレータをVDD=3.3V出力で動作させる場合(OCD)  E20、IE850、IE850Aエミュレータではサポートしない |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_EXECRUN EXECは既に起動している

EX\_FE\_ANOTHEREXECRUN ディレクトリが異なる別のEXECが既に動作している

EX\_FE\_DEVFILE デバイスファイルがない、または、読み込みに失敗した

EX\_EE\_BUFFAILD パソコンのメモリの取得に失敗

EX\_FE\_COMM デバイスドライバのオープンに失敗した

EX\_SE\_M\_EMUACCFAILED エミュレータへのアクセスに失敗

EX\_SE\_TRGTOFF ターゲットパワーOFF

EX\_SE\_NS\_TPON E1エミュレータから電源を供給する指定なのに既にターゲット電源が入っている

EX\_NOSUPRT サポートしていない機能を指定した

EX\_FE\_TGTNONCNCT ターゲットボードが接続されていない

EX\_FE\_ICENUMOVER PCに接続されたICE数が規定値をオーバー

EX\_UE\_NOSPRT\_IE850 指定したデバイスはIE850を未サポート

EX\_UE\_NOSPRT\_IE850A　　　指定したデバイスはIE850Aを未サポート

EX\_SE\_NOCONNECT\_EMUADPT エミュレーションアダプタが接続されていない

EX\_SE\_CONNECT\_IE850 IE850エミュレータが接続されている

(注)以下は主にデバイスとのLPD通信を開始する際に発生するエラー。(場合によっては通常の関数でも通信不良の時には発生する可能性もある)。なお、OCDの場合は本関数ではLPD通信開始処理を行わないため発生しない。FULLエミュレータではこの関数でLPD通信開始処理までおこなうかもしれないので残してある。

EX\_DE\_M\_DCUACCESS DCUアクセス異常

EX\_DE\_M\_MONACCFAILED モニタエリアアクセス失敗

EX\_UE\_M\_BRKRSTFAILED ブレークリセット失敗

EX\_DE\_M\_MONEXEFAILED モニタ実行失敗

[Description]

[引数説明]

hWndはNULL固定とする。

devfile\_infは省略できない、デバイスファイルがない場合は、エラーコード(EX\_FE\_DEVFILE)を返す。

user\_inf.vdd\_mode= EX\_VDD\_5V\_OUT、EX\_VDD\_3V\_OUTは、E1、E2エミュレータのターゲット電源供給機能の指定となる。E1、E2エミュレータのターゲット電源供給機能が利用できるよう設計されているターゲット接続の時のみ設定すること。設定がない場合はターゲットに電源がある前提で起動する。

user\_inf.vdd\_mode=EX\_VDD\_TARGET\_HAとすることで、ホットプラグイン起動を行う。

IE850Aではuser\_inf.vdd\_modeの値を無視し、IE850AからのVOUT0/1の電源供給を行う。

未使用パラメータは、将来使うかもしれないのでゼロクリアしておくこと。

[動作説明]

EXECの起動を行う。

この関数が正常終了した場合は、終了時にex\_exit()を呼び出すこと。

ターゲット電源のチェックまで行う。ターゲット接続状態で、10秒待ってもターゲット電源が入らない場合は、EX\_SE\_TRGTOFFエラーを返す。

OCDの場合、その後のLPD通信開始処理はOCDID(セキュリティ用ID)解除処理(ex\_socunitinfo2(EX\_UNIT\_RSU))にて行う。またその前に、LPDクロック等の設定は別関数(ex\_socunitinfo2(EX\_UNIT\_LPDOPT))で設定する。

接続するエミュレータがIE850であるが、指定したデバイスファイルがIE850をサポートしていない品種の場合は、エラー(EX\_UE\_NOSPRT\_IE850)を返す。

同様に、接続エミュレータがIE850Aであるが、指定したデバイスファイルがIE850Aをサポートしていない品種の場合は、エラー(EX\_UE\_NOSPRT\_IE850A)を返す。

user\_inf.vdd\_modeにEX\_VDD\_NORMALを指定した場合、E1/E20/E2/IE850エミュレータは電源を供給せずにEXECを起動する。IE850AはVOUT0に5.5[V]、VOUT1に3.3[V]を供給状態でEXECを起動する。

user\_inf.vdd\_modeにEX\_VDD\_5V\_OUTまたはEX\_VDD\_3V\_OUTを指定した場合、以下の動作となる。

【E1、E2エミュレータ接続時】

5Vまたは3.3Vの電源をターゲットに供給する。この時、ターゲット電源が入っている場合は、EX\_SE\_NS\_TPONエラーを返して電源を供給しない。

【E20エミュレータ接続時】

EX\_NOSUPRTエラーを返して電源を供給しない。

【IE850、IE850Aエミュレータ接続時】

指定が無視され、EX\_VDD\_NORMALを指定した場合と同じ動作となる。

E2エミュレータでは、user\_inf.vdd\_modeの指定が有効となるのは、本関数の呼び出し前にex\_extpower()が呼ばれなかった場合のみ。本関数の呼び出し前にex\_extpower()が呼ばれた場合は、user\_inf.vdd\_modeの指定は無視する。下記のホットプラグイン接続指定についても同様。

user\_inf.vdd\_mode=EX\_VDD\_TARGET\_HAとすることで、ホットプラグイン起動を行う。ホットプラグイン

アダプタにE1/E20/E2ユーザインターフェースケーブルを未接続の状態で呼び出すこと。正常終了後、ユーザにユーザインターフェースケーブルを接続してもらう。その後、EXEC関数ex\_socunitinfo2()unit\_type=EX\_UNIT\_RSUを呼び出すと、ホットプラグインRUN状態で接続する。ex\_initexec()呼び出し後は、ホットプラグイン中断する際でも、ex\_exit()を呼び出すこと。~~ホットプラグインRUN中にターゲット電源OFF→ONした場合は強制ブレークとなる。~~(取り消し線部分は未サポート機能)

IE850エミュレータ接続時にホットプラグイン起動を指定した場合、エラー(EX\_SE\_CONNECT\_IE850)を返す。

IE850A接続時にホットプラグイン起動を指定した場合、何もせずエラーなし(EX\_NOERROR)を返す。

ホットプラグインRUN中(接続～強制ブレークするまで)に通常動作と異なる関数は以下の通り。

異なる動作の詳細については各関数のDescriptionを参照。

・ex\_setenv()

・ex\_memread()

・ex\_memwrite()

・ex\_memfill()

・ex\_regread()

・ex\_regwrite()

・ex\_rh\_evncond()

・ex\_rh\_evnintgcond()

・ex\_hwbrkevent()

・ex\_swbrkcond()

・ex\_trcrun()

・ex\_trcmode()

・ex\_trcclear()

・ex\_trcevent()

・ex\_trcdelay()

・ex\_rh\_trceventsel2()

・ex\_rh\_trcbufcont()

・ex\_trcrawmeminfo()

・ex\_trcrawmemread()

・ex\_rh\_trchwevncopeinfo()

・ex\_timeevent2()

・ex\_rh\_timeevent()

・ex\_pincont()

・ex\_socunitinfo2()

・unit\_type = EX\_UNIT\_RSU

・unit\_type = EX\_UNIT\_CG

・unit\_type = EX\_UNIT\_FLFUNCINFO

・unit\_type = EX\_UNIT\_FLMAPINFO

・unit\_type = EX\_UNIT\_FLBLOCKINFO

・unit\_type = EX\_UNIT\_DFLNUMINFO

・unit\_type = EX\_UNIT\_DFLBLOCKINFO

[デバッガ関係者向け注意事項]

E20エミュレータまたはIE850を使用している場合は、ex\_initexecコマンドによるターゲット電源の供給をサポートしない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_reset | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

CPUのリセットを行う

[Format]

int ex\_reset( USHORT mode )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | リセットのモード | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | EX\_RESET | ユーザプログラム実行中であればリセット後、ブレークする。 |
|  | EX\_NONBR\_RES | ユーザプログラム実行中であればリセット後、リセットベクタからユーザプログラムを継続実行する。(ユーザ非公開) |
|  | EX\_M\_RES\_CONT | リセットを継続する。(ユーザ非公開) |
|  | EX\_F\_RESET | リセット入力後、ブレークする。 |
|  | EX\_INITSTOP\_BRKRES | リセット入力後、全コアを起動してブレークする。（ユーザ非公開、初期停止・スタンバイモードデバッグ環境が有効時のみ使用可能) |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_UE\_RESET リセット継続中でリセット解除できない

EX\_UE\_M\_BRKRSTFAILED ブレークリセット失敗

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_NOSPRT\_DURING\_CANBOOT CANブートデバッグ中のため本関数は使用できない

EX\_SE\_HOTPLUG\_STANDBY\_NOSPRT ホットプラグイン接続後の該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態で操作してください。

[Description]

[引数説明]

通常はブレーク中でmodeにEX\_RESETを指定して発行すること。

デバッガでユーザプログラム実行中のCPUリセットを実装する場合には、まず、ex\_stop()でブレークさせてからmodeにEX\_RESETを指定してex\_reset()を使うこと。

ユーザプログラム実行中のex\_stop()発行で下記エラーが発生した場合は、非常手段としてmodeにEX\_F\_RESETを指定したex\_reset()を発行可能である。

modeにEX\_F\_RESETを指定したex\_reset()を利用可能なex\_stop()発行時のエラー：

・EX\_UE\_ENFBRK (強制ブレークできない)

・EX\_SE\_FETCHSTOP\_BEFORE\_BRKFAILED(初期停止中のためブレーク要求を出さないで処理を中断した。)

EX\_NONBR\_RESとEX\_M\_RES\_CONTは、EXECデバッグ用とする。

EX\_INITSTOP\_BRKRESは、初期停止・スタンバイモードデバッグ環境が有効(ex\_socunitinfo2のunit\_type=EX\_UNIT\_INITSTOPDBG)の場合のみ使用可能とする。(デバッガおよびEXECでのみ使用可能でユーザには非公開)

[動作説明]

CPUリセットを行う。

ユーザプログラム実行中にいきなりex\_reset(mode=EX\_RESET)でリセットブレークした場合、EXEC内部処理でのブレークの後処理が不完全に終わる可能性があるので、ブレーク中に利用すること。

modeのEX\_F\_RESETは実行中でも利用可能である。発行するとCPUリセットを行ない、全てのCPUをユーザプログラム停止中に遷移させる。ただし、デバイスの状態によっては、正常に終了することができず、その後デバッグの継続ができない場合がある。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

G4世代MCUの場合、エミュレータ未接続時は初期値が不定である以下のレジスタを、エミュレータ接続時は起動処理完了時に初期化する(不定状態でのレジスタ読み出しによるECMロックステップエラーを抑止するため)処理を行なうが、CPUリセット発行時にも同様に初期化する。なお、レジスタの初期化はロックステップ搭載コアのみ行ない、ロックステップ未搭載コアは行わない。

初期化するレジスタ：R1、R3、R[31:6]、EIPC、デバッグ用レジスタ

初期化する値　　　：0x00000000

CANブートデバッグ時に本コマンドを実行すると、エラー(EX\_SE\_NOSPRT\_DURING\_CANBOOT)が返る。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・ex\_socunitinfo2のunit\_type=EX\_UNIT\_INITSTOPDBGで初期停止・スタンバイモードデバッグ環境を有効に設定し、ex\_resetコマンドで(mode=EX\_RESET)または(mode=EX\_F\_RESET)を指定した場合、一度全コアを起動させるブレークリセット処理を行い、停止(初期停止・Cyclic)中のコアに行えないCPUアクセスを伴うCPUコア(実行前)イベントブレーク設定などを削除する処理を行った後、初期停止状態を保持したブレークリセットを発行してユーザプログラム停止中に遷移させる。

・ex\_socunitinfo2のunit\_type=EX\_UNIT\_INITSTOPDBGを初期停止・スタンバイモードデバッグ環境を有効に設定した場合のみ、ex\_resetコマンドで(mode=EX\_INITSTOP\_BRKRES)を指定可能である。

動作としては、全てのコアを起動させるブレークリセットを発行してユーザプログラム停止中の状態とする。なお、デバッガおよびEXECでのみ使用可能としユーザには非公開とする。

・初期停止・スタンバイモードデバッグ環境を無効に設定して、ex\_resetコマンドで(mode=EX\_INITSTOP\_BRKRES)を指定した場合、EX\_PE\_DATAのエラーを返す。

・ホットプラグイン接続後、スタンバイモード中にex\_resetコマンドを発行した場合、エラー(EX\_SE\_HOTPLUG\_STANDBY\_NOSPRT)応答する。

[マルチコア時の注意点]

PE選択状態に関係なく、すべてのPEをリセットする。

modeにEX\_F\_RESETを指定した場合、初期停止状態であったCPUは初期停止状態を解除してユーザプログラム停止中に遷移する。

[非同期デバッグモード時の注意点]

非同期デバッグモード時もPE選択状態に関係なく、すべてのPEをリセットする。

modeにEX\_F\_RESETを指定した場合、初期停止状態であったCPUは初期停止状態を解除してユーザプログラム停止中に遷移する。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_setemulator | **SYNC** | **EMU** | **×** | **BRK** | **×** | **TRC** | **×** | **TIM** | **×** | **COV** | **／** | **HOT** | **×** |
| **※本関数は起動前しか呼べないため、同期/非同期は関係しない** | | | | | | | | | | | | |

[Function]

接続するエミュレータの種類やシリアル文字列、または、任意でエミュレータ個別に与えたニックネーム文字列を指定する

[Format]

int ex\_setemulator( UCHAR \*sid, ULONG drvtype, ULONG flag, ULONG \*stat )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| UCHAR | \*sid | <I > | シリアル文字列指定 | ○ |
| ULONG | drvtype | <I > | エミュレータ種別指定 | ○ |
| ULONG | flag | <I > | オプション | ○ |
| ULONG | \*stat | < O> | エミュレータ情報 | ○ |

[Arguments]

|  |  |  |  |
| --- | --- | --- | --- |
| sid | | | |
|  | エミュレータのシリアル文字列、または、ニックネーム文字列 | | |
|  |  | | |
| drvtype | | | |
|  | エミュレータ種別 | | |
|  | EX\_DRVTYPE\_ALL | | 指定なし |
|  | EX\_DRVTYPE\_E1USB | | E1 |
|  | EX\_DRVTYPE\_E20USB | | E20 |
|  | EX\_DRVTYPE\_E2USB | | E2 |
|  | EX\_DRVTYPE\_IECUBE2\_V850 | | IE850 |
|  | EX\_DRVTYPE\_IE850AUSB | | IE850A |
|  |  | |  |
|  | (以下は参考) | |  |
|  | EX\_DRVTYPE\_EZCUBE | | EZ-CUBE |
|  | EX\_DRVTYPE\_MINICUBE\_78K0 | | 78K0-MINI |
|  | EX\_DRVTYPE\_MINICUBE\_78K0S | | 78K0S-MINI |
|  | EX\_DRVTYPE\_MINICUBE\_V850 | | V850-MINI(L) |
|  | EX\_DRVTYPE\_IECUBE\_V850 | | V850-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0 | | 78K0-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0S | | 78K0S-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0R | | 78K0R-IECUBE |
|  | EX\_DRVTYPE\_MINICUBE2 | | MINICUBE2 |
|  | EX\_DRVTYPE\_E2LITEUSB | | E2 Lite |
|  |  | |  |
| flag | | | |
|  | エミュレータ検索方法の拡張オプション | | |
|  | bit0 | EX\_FLAG\_PARTIALMATCH | 0：シリアル文字列を完全一致で比較  1：シリアル文字列を部分一致で比較 |
|  | bit1-30 | | 未使用(0固定) |
|  | bit31 | EX\_FLAG\_NICKNAME | 0：シリアル文字列を検索する  1：ニックネームを検索する |
|  |  | |  |
| stat | | | |
|  | 指定エミュレータの接続状態を返す | | |
|  | EX\_ICESTAT\_NOTFOUND | | 指定エミュレータが見つからない |
|  | EX\_ICESTAT\_NOTAVAILABLE | | 指定したエミュレータは他のEXECが使用中 |
|  | EX\_ICESTAT\_AVAILABLE | | 指定したエミュレータは接続可能 |
|  | EX\_ICESTAT\_ICENUMOVER | | 接続されたICE数が規定値オーバー |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_EXECRUN 既にドライバがオープンされている

[Description]

[引数説明]

(1)sid

接続対象エミュレータのシリアル文字列、または、ニックネーム文字列を指定する(最大259文字まで。260文字以上は指定しないこと)。シリアル文字列を検索するかニックネーム文字列を検索するかはflagの指定に依存する。

シリアル文字列は大きく分けて、E1、E20、E2、IE850Aなどのシリアル番号がエミュレータに埋め込まれているタイプとIE850のように埋め込まれていないタイプとが存在する。

シリアル番号が埋め込まれているタイプはエミュレータの背面に貼られたシールに記載の番号がXXXXXXXXXの場合、「E1:\_XXXXXXXXX」または「E20:\_XXXXXXXXX」、「E2:\_XXXXXXXXX」、「IE850A:\_ XXXXXXXXX」のように指定する。シールがない場合でも、デバイスインスタンスパス(WinXPではデバイスインスタンスID、以下省略)を参照することでシリアル文字列が分かる。

シリアル番号が埋め込まれていないタイプのエミュレータはデバイスインスタンスパスを参照することでしかシリアル文字列を参照できない。このタイプのデバイスインスタンスパスは接続するホストPCや接続するUSBポートが異なると割り当てられる文字列が変わるため、環境が変わった場合は必ず参照しなおすこと。

以下にデバイスインスタンスパスの例を示す。本引数には下線部の文字列を指定する。文字列に含まれるアルファベッドの大文字と小文字は区別しないため、どちらで引数に渡しても構わない。

・E1 USB\VID\_045B&PID\_823B\E1:\_9JM000109

・IE850 USB\VID\_0409&PID\_01D8\6&49703C3&0&1

エミュレータのシリアル文字列を指定しない場合はNULLを指定すること。

NULLを指定した場合は、最初に見つかったエミュレータに対して接続を試みる。

もし、最初に見つかったエミュレータが接続中であった場合は、他のエミュレータがPCに接続されていてもstat=EX\_ICESTAT\_NOTAVAILABLEを返す。

ニックネーム文字列はユーザが任意でIE850のFW、E2のFWに書き込んだ文字列である。書き込みはIE850用のユーティリティソフト、E2用の自己診断ソフトにて行う。ただし、ニックネーム機能は特定顧客向けのカスタム機能である。

(2)drvtype

接続対象のエミュレータの種類を指定する。

いずれか1種類のエミュレータ種類のみ指定可能。複数のエミュレータ種類を指定してもエラーは返さないが、検索結果に失敗が返る。また、RH850でサポートするエミュレータ種類(E1、E20、E2、IE850、IE850A)以外の種別を指定してもエラーにならないが、検索結果に失敗が返る。

EX\_DRVTYPE\_ALLを指定するとエミュレータの種類指定なしにできる。

(3)flag

ビット単位でエミュレータ検索のオプションを指定できる。

bit0： シリアル文字列検索オプション

0：完全一致

1：部分一致

bit31: ニックネーム検索オプション

0：シリアル文字列を検索

1：ニックネーム文字列を検索

(4)stat

検索対象となるエミュレータの検索結果および接続状況が返る。

[動作説明]

接続するエミュレータの検索対象を指定する。

本I/Fを実行せずに任意のEXEC I/F関数を呼び出した場合(従来の接続手順)、次の順番でホストPCに接続されているエミュレータを検索し、最初に見つかったエミュレータに対して接続を行う。

(1)IE850

(2)E1、E20(E1とE20の区別はしていない)

(3)E2、IE850A(E2とIE850Aの区別はしていない)

本I/Fで接続するエミュレータのシリアル文字列やエミュレータの種類を指定することで、最初に見つかったエミュレータ以外のエミュレータに対しても接続が可能となる。

本I/Fは接続対象のエミュレータを検索するだけで実際にエミュレータに対する接続処理は行わない。本I/F以外のEXEC I/Fを実行すると、エミュレータへの接続処理が行われるため、本I/Fを最初に実行する必要がある。他のEXEC I/Fを実行後に本I/Fを実行すると、エラー(EX\_FE\_EXECRUN)が返る。

エミュレータの種類とシリアル文字列の検索条件はANDとなる。下記の2つのエミュレータがホストPCに接続された状態で  
drvtype=EX\_DRVTYPE\_IECUBE2\_V850、sid="E1:\_9JM000109"で検索した場合、発見されないことになる。

・E1 USB\VID\_045B&PID\_823B\E1:\_9JM000109

・IE850 USB\VID\_0409&PID\_01D8\6&49703C3&0&1

シリアル文字列検索のオプションとして完全一致と部分一致を選択可能。デフォルトではシリアル文字列の完全一致で検索するが、部分一致を指定すると上記の「E1:\_9JM000109」のエミュレータの検索は  
「9JM000109」など指定したシリアル文字列がデバイスインスタンスパスの一部に含まれるエミュレータを検索することができる。

指定したシリアル文字列のエミュレータが発見されなかった場合はstatにEX\_ICESTAT\_NOTFOUNDが返り接続できない。指定したエミュレータが発見されても、他のEXECが既に指定したエミュレータに接続している場合はstatにEX\_ICESTAT\_NOTAVAILABLEが返り、この場合も接続できない。接続可能な場合はstatにEX\_ICESTAT\_AVAILABLEが返る。シリアル文字列もエミュレータ種別も指定しなかった場合は、従来と同じく最初に見つかったエミュレータに接続可能かどうかがstatに返る。このとき、エミュレータが何も接続されていない場合は、statにEX\_ICESTAT\_NOTFOUNDが返る。PCに接続可能なエミュレータの台数(E1/E20で合計4台まで、E2/IE850Aは4台まで、IE850は4台まで)を超えて接続していた場合はstatにEX\_ICESTAT\_ICENUMOVERが返る。

[ニックネーム機能について]

ニックネーム機能はE2/IE850AおよびIE850のみの機能であり、E1/E20はサポートしていない。

ニックネーム機能を使用する場合、予めE2は自己診断ソフト、IE850は専用ユーティリティソフトにてニックネーム文字列を書き込んでおく必要がある。

ニックネーム文字列を検索する際は、flagでEX\_FLAG\_NICKNAMEを1にする。ニックネーム機能では部分一致オプションは使用できない。指定しても無視する。

ニックネーム文字列の検索はシリアル文字列の検索とは異なり、実際にドライバ接続してE2/IE850A、IE850から文字列を読み出す必要があるため、既に別のEXECで接続されているE2/IE850A、IE850に対しては、ニックネームの有無や一致しているかどうかを確認することもできない。そのため、検索対象のニックネームが書かれたE2/IE850A、IE850があったとしても、そのE2/IE850A、IE850が既に別のEXECで接続済みの場合は、見つからなかったとしてstatにEX\_ICESTAT\_NOTFOUNDが返る。

[新旧ドライバについて]

IE850用の32bitUSBドライバはV850E2でも使用していた旧ドライバとThesycon社製の新ドライバ(2014年リリース予定)が存在することになる。64bitUSBドライバは既にThesycon社製のドライバを使用しており、32bitのドライバが新ドライバに置き換わってもEXECで特別な対応は必要なく動作させることが可能。新ドライバはインストール可能なOSを限定する方式にはせず、すべてのOSでインストール可能とするため、すべてのOSで新旧ドライバ混在の可能性がある。Win7以降はOSが新ドライバを判定して接続されるため、意識せずに新ドライバで接続される。XP、Vistaではインストールの状況により旧ドライバで接続される可能性があるが、旧ドライバで接続した場合の問題※が発生しないため、旧ドライバで接続されても正常に動作する。

※Windows7と旧ドライバの組み合わせでUSBセレクティブサスペンドを有効に設定した場合、サスペンド状態になったUSBに接続されているエミュレータとの接続再開ができなくなる問題。現状は「USBセレクティブサスペンドを有効にしないでください」という回避策としている。

[デバッガ関係者向け注意事項]

IE850はV850E2と同じドライバを使用していることにより、F/WやFPGAをRH850用に書き換えていないIE850も接続対象となってしまう。V850E2用のEXECの初期の版では複数台接続の排他制御が行われていないものもあり、一つのエミュレータに対してV850E2のEXECとRH850のEXECが接続しようとした場合にエラーが返らずに処理をしてしまう場合がある。この場合、ホストPCの電源が落ちる危険性がある。複数台接続をする場合はV850E2用EXEC、RH850用EXECともに複数台接続に対応した版を使用すること。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_getemulist | **SYNC** | **EMU** | **×** | **BRK** | **×** | **TRC** | **×** | **TIM** | **×** | **COV** | **／** | **HOT** | **×** |
| **※本関数は起動前しか呼べないため、同期/非同期は関係しない** | | | | | | | | | | | | |

[Function]

PCに接続されているエミュレータのリストを取得する

[Format]

int ex\_getemulist( ULONG drvtype, ULONG listnum, ULONG \*emulistnum, EXEMULIST \*emulist )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | drvtype | <I > | エミュレータ種別指定 | ○ |
| ULONG | listnum | <I > | 取得するリスト数 | ○ |
| ULONG | \*emulistnum | < O> | 実際に接続されているエミュレータ数 | ○ |
| EXEMULIST | \*emulist | < O> | エミュレータリスト | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | sid[32] | < O> | シリアル文字列 | ○ |
|  | UCHAR | nid[201] | < O> | ニックネーム文字列 | ○ |
|  | ULONG | stat | < O> | 接続状況 | ○ |
| }EXEMULIST | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| drvtype | | |
|  | エミュレータ種別 | |
|  | EX\_DRVTYPE\_E1USB | E1 |
|  | EX\_DRVTYPE\_E20USB | E20 |
|  | EX\_DRVTYPE\_E2USB | E2 |
|  | EX\_DRVTYPE\_IE850AUSB | IE850A |
|  |  |  |
|  | (以下は参考) |  |
|  | EX\_DRVTYPE\_ALL | 指定なし |
|  | EX\_DRVTYPE\_EZCUBE | EZ-CUBE |
|  | EX\_DRVTYPE\_MINICUBE\_78K0 | 78K0-MINI |
|  | EX\_DRVTYPE\_MINICUBE\_78K0S | 78K0S-MINI |
|  | EX\_DRVTYPE\_MINICUBE\_V850 | V850-MINI(L) |
|  | EX\_DRVTYPE\_IECUBE\_V850 | V850-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0 | 78K0-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0S | 78K0S-IECUBE |
|  | EX\_DRVTYPE\_IECUBE\_78K0R | 78K0R-IECUBE |
|  | EX\_DRVTYPE\_MINICUBE2 | MINICUBE2 |
|  | EX\_DRVTYPE\_IECUBE2\_V850 | IE850 |
|  | EX\_DRVTYPE\_E2LITEUSB | E2 Lite |
|  |  |  |
| listnum | | |
|  | <00000001h-00000004h> |  |
|  |  |  |
| emulistnum | | |
|  |  | PCに接続されているエミュレータ数 |
|  |  |  |
| sid |  |  |
|  |  | PCに接続されているエミュレータのシリアル文字列 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| nid |  |  |
|  |  | PCに接続されているエミュレータのニックネーム文字列  (本開発フェーズではサポートしない) |
|  |  |  |
| stat |  |  |
|  | EX\_EMULIST\_NOTUSE | エミュレータは未使用 |
|  | EX\_EMULIST\_INUSED | エミュレータは使用中 |

[Return Values]

EX\_NOERROR 正常終了

EX\_PE\_DATA 設定値が異常

EX\_FE\_EXECRUN 既にドライバがオープンされている

EX\_FE\_COMMUNI\_ERROR Communi.dllファイルが見つからない

[Description]

[引数説明]

(1)drvtype

取得するエミュレータの種類を指定する。

いずれか1種類のエミュレータ種類のみ指定可能。複数のエミュレータ種類を指定するとエラー(EX\_NOERROR)が返る。また、サポートするエミュレータ種類はE1、E20、E2、IE850Aのみで、他のエミュレータを指定するとエラー(EX\_PE\_DATA)が返る。EX\_DRVTYPE\_ALLもエラー(EX\_PE\_DATA)が返る。EX\_DRVTYPE\_IECUBE2\_V850もIE850がシリアルIDを持っていないためエラー(EX\_PE\_DATA)とする。

(2)listnum

取得するエミュレータリスト数を指定する。1～4以外指定した場合はエラー(EX\_PE\_DATA)を返す。

(3)emulistnum

PCに接続されているエミュレータの台数を返す。ただし、listnum以下の値を返す。5台以上のエミュレータがPCに接続されていても最大4台分のエミュレータリストしか返さない。

(4)sid

取得したエミュレータのシリアル文字列が返る。

(5)nid

取得したエミュレータのニックネーム文字列が返る(本開発バージョンではサポートしない)。

(6)stat

取得したエミュレータの使用状況が返る。

[動作説明]

PCに接続されているエミュレータのシリアル文字列(emulist.sid)と使用状況(emulist.stat)を取得する。

「**2.2.1起動**」の起動シーケンスで示したようにリスト取得する場合は最初に呼び出すこと。

デバッガは取得するエミュレータの検索対象(drvtype)と取得するエミュレータリスト数(listnum)を指定する。指定するエミュレータリスト数(listnum)は取得するエミュレータリストの最大数であり、実際に接続されているエミュレータの数が少ない場合は接続されている台数分のリスト(emulist)しか返らない。接続されていたエミュレータの台数はemulistnumに返る。リストの最大数は4台で5台以上接続されていても4台分までのリスト(emulist)しか返さない。

現在のプロセス上でロードしているEXECが既にいずれかのエミュレータに接続している状態で呼び出した場合はエラー(EX\_FE\_EXECRUN)が返る。他のプロセス上でロードしているEXECが接続していてもエラーにはならず、取得したエミュレータリストのstatがEX\_EMULIST\_INUSEDとなるだけである。

Communi.dllがロードできなかった場合はエラー(EX\_FE\_COMMUNI\_ERROR)が返る。

[デバッガ関係者向け注意事項]

(1) RFWのI/F仕様に合わせてlistnumパラメータを用意しているが、listnumは4を指定すること。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_extpower | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

電源供給設定を行う

[Format]

int ex\_extpower( USHORT mode, USHORT val )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 電源供給のモード | ○ |
| UCHAR | val | <I > | 供給する電圧値 | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
| 【E2エミュレータの場合】 | | |
|  | EX\_EXTPWR\_TARGET | 電源供給しないで通常接続する。 |
|  | EX\_EXTPWR\_TARGET\_ASPIF | 電源供給しないで通常接続する。AssistProbe(ASP) I/Fをターゲット電源で使用する。 |
|  | EX\_EXTPWR\_HOTPLUG | 電源供給しないでホットプラグイン接続する。 |
|  | EX\_EXTPWR\_HOTPLUG\_ASPIF | 電源供給しないでホットプラグイン接続する。ASP I/Fをターゲット電源で使用する。 |
|  | EX\_EXTPWR\_USERIF | ユーザI/Fのみに電源供給する。 |
|  | EX\_EXTPWR\_ASPIF | ASP I/Fのみ電源供給する。 |
|  | EX\_EXTPWR\_USERIF\_ASPIF | ユーザI/FとASP I/F両方に電源供給する。 |
| 【IE850Aエミュレータの場合】 | | |
|  | EX\_EXTPWR\_TARGET | 電源供給しないで通常接続する。 |
|  | EX\_EXTPWR\_TARGET\_ASPIF | IE850Aは未サポート |
|  | EX\_EXTPWR\_HOTPLUG | 電源供給しないでホットプラグイン接続する。 |
|  | EX\_EXTPWR\_HOTPLUG\_ASPIF | IE850Aは未サポート |
|  | EX\_EXTPWR\_VOUT0 | VOUT0のみに電源供給する。 |
|  | EX\_EXTPWR\_VOUT1 | VOUT1のみに電源供給する。 |
|  | EX\_EXTPWR\_VOUT0\_VOUT1 | VOUT0とVOUT1両方に電源供給する。 |
|  |  |  |
| val |  |  |
| 【E2エミュレータの場合】 | | |
|  | <0x00、0x12-0x32> | 0.1V単位で供給する電圧を指定する。供給する電圧の10倍した値を入力する。指定可能範囲：0V、1.8V～5.0V  ※RHではvalに0x0を指定しないこと。 |
| 【IE850Aエミュレータの場合】 | | |
|  | <0x00、0x0C-0x37> | 0.1V単位でVOUT0に供給する電圧を指定する。供給する電圧の10倍した値を入力する。指定可能範囲：0V、1.2V～5.5V  ※1 RHではvalに0x0を指定しないこと。  ※2 VOUT1は値に関わらず3.3Vを供給する。  ただし、電圧は1.2V～5.5Vを指定すること。 |

[Return Values]

EX\_NOERROR 正常終了

EX\_PE\_DATA 設定値が異常

EX\_NOSUPRT 本関数は未サポート

EX\_SE\_NS\_TPON エミュレータから電源を供給する指定なのに既にターゲット電源が入っている

EX\_FE\_TGTNONCNCT ターゲットボードが接続されていない

EX\_SE\_PWRMODE\_NOCHG 接続後の電源供給I/Fは変更できない

EX\_FE\_VOUTERROR IE850AがVOUTからの電源供給で過電流を検出しました

EX\_FE\_BEFOREVOUTERROR IE850AがVOUTから電源供給を開始できませんでした

EX\_SE\_CONNECT\_IE850A IE850Aエミュレータが接続されている

[Description]

[引数説明]

mode：電源供給の有無、電源供給先を選択する。ホットプラグイン接続の場合は電源供給できないためその指定も本関数で行う。

val ：電源供給する場合の電圧値を指定する。指定単位は0.1V単位となる。接続エミュレータによって以下の動作となる。

【E2エミュレータ接続時】

電源供給する対象は、ターゲットシステム(USER I/F)と拡張機能(ASP I/F)となる。供給電圧を0Vと1.8～5.0Vの範囲で指定が可能。

【IE850Aエミュレータ接続時】

電源供給する対象は、エミュレーションアダプタとなるがエミュレーションアダプタ以外を接続中においても、電源供給は行われる。VOUT0に対しての供給電圧を1.2V～5.5Vの範囲で指定が可能。

なお、VOUT1の供給電圧は3.3V固定値となり指定することはできないが、指定供給電圧には1.2V～5.5Vの値を指定すること。

E2エミュレータ/IE850Aエミュレータ接続時共に、0を指定すると電源を供給するmodeを指定しても電源は供給されない。そのため、EX\_EXTPWR\_TARGETなどのmodeと同じ動作となるためvalに0は指定せず、供給しないmodeを指定すること。valの0指定はターゲット接続後の電圧変更でターゲット電源遮断状態のエミュレーションを目的としているが、RH EXECのOCDでは電源遮断エミュレーションをサポートしていないため、valに0は指定しないこと。

E2/IE850Aエミュレータでのmodeパラメータの違いを以下の表にまとめる。

表 3‑2　ex\_extpowerのmodeパラメータ一覧【E2エミュレータの場合】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mode | 起動方法 | 電源供給有無 | | ASP I/Fの使用可否 |
| USER I/F | ASP I/F |
| EX\_EXTPWR\_TARGET | 通常起動 | 供給しない | 供給しない | 不可 |
| EX\_EXTPWR\_TARGET\_ASPIF | 通常起動 | 供給しない | 供給しない | 可(ターゲット電源) |
| EX\_EXTPWR\_HOTPLUG | ホットプ  ラグイン | 供給しない | 供給しない | 不可 |
| EX\_EXTPWR\_HOTPLUG\_ASPIF | ホットプ  ラグイン | 供給しない | 供給しない | 可(ターゲット電源) |
| EX\_EXTPWR\_USERIF | 通常起動 | 供給する | 供給しない | 不可 |
| EX\_EXTPWR\_ASPIF | 通常起動 | 供給しない | 供給する | 可(エミュレータからの  供給電源) |
| EX\_EXTPWR\_USERIF\_ASPIF | 通常起動 | 供給する | 供給する | 可(エミュレータからの  供給電源) |

表 3‑3　ex\_extpowerのmodeパラメータ一覧 【IE850Aエミュレータの場合】

|  |  |  |  |
| --- | --- | --- | --- |
| mode | 起動方法 | 電源供給有無 | |
| VOUT0 | VOUT1 |
| EX\_EXTPWR\_TARGET | 通常起動 | 供給しない | |
| EX\_EXTPWR\_TARGET\_ASPIF | 指定不可 | | |
| EX\_EXTPWR\_HOTPLUG | 指定不可 | | |
| EX\_EXTPWR\_HOTPLUG\_ASPIF | 指定不可 | | |
| EX\_EXTPWR\_VOUT0 | 通常起動 | 供給する | 供給しない |
| EX\_EXTPWR\_VOUT1 | 通常起動 | 供給しない | 供給する |
| EX\_EXTPWR\_VOUT0\_VOUT1 | 通常起動 | 供給する | |

[動作説明]

E2エミュレータ/IE850Aエミュレータの電源供給設定を行う。E2エミュレータ/IE850Aエミュレータ以外で、本関数を呼び出した場合はEX\_NOSUPRTエラーが返る。電源供給する設定でvalに範囲外の値を設定した場合はEX\_PE\_DATAエラーが返る。

ex\_initexec()でも電源供給設定が可能だが、ex\_initexec()関数実行前に本関数を呼び出すと、ex\_initexec()関数での電源供給設定は無視する。

本関数は、ターゲットへの接続完了後にも呼び出しが可能(ユーザプログラム実行中でも可)。接続後でもvalの指定値を変更することが可能。modeの指定値は接続後の変更は不可。接続前のmodeと違う値を指定した場合はEX\_SE\_PWRMODE\_NOCHGエラーが返る。modeの初期値はE2エミュレータでEX\_EXTPWR\_TARGET、IE850AでEX\_EXTPWR\_VOUT0\_VOUT1である。ターゲット接続後に本関数で供給電圧を変えたい場合は、ex\_initexec()より前に、本関数を呼び出しておく必要がある。本関数を呼び出さず、ex\_initexec()で接続した場合は、本関数での設定値は無視する(特にエラーは返さない)。

ターゲットボードが未接続の場合はEX\_FE\_TGTNONCNCTエラーが返る。

【E2エミュレータ接続時】

本関数を呼び出した時点で電源供給が開始される。ex\_exit()関数で電源供給が止まる。

電源供給指定されたが、既にターゲット電源が入っている場合はEX\_SE\_NS\_TPONエラーが返る。

【IE850Aエミュレータ接続時】

エミュレータの電源をONにした時から電源供給を行う仕様のため、本関数を呼び出した時点で既に電源供給が行われている。従って、既にターゲット電源が入っている場合のエラー(EX\_SE\_NS\_TPON)は返さない。また、ex\_exit()関数を発行した時、VOUT0に5.5V供給、VOUT1に3.3V供給の初期状態となる。

IE850Aエミュレータ接続時、modeに「IE850Aは未サポート」の値を設定した場合は、エラー(EX\_SE\_CONNECT\_IE850A)を返す。

IE850Aエミュレータからの供給電圧が異常値となっていた場合、エラー(EX\_FE\_VOUTERROR)を返す。供給電圧の異常を検出し、IE850AエミュレータのVCCLEDが消灯している状態では、IE850Aエミュレータは電源供給を停止し、電源供給設定の変更も受け付けない。

IE850Aエミュレータの電源をONにした時、VOUT0またはVOUT1に0.5V以上の電圧が検出された場合、エラー(EX\_FE\_BEFOREVOUTERROR)を返す。この場合も供給電圧の異常と判断し、IE850AエミュレータのVCCLEDが消灯し、IE850Aエミュレータは電源供給を停止し、電源供給設定の変更も受け付けない。

[デバッガ関係者向け注意事項]

(1) 本関数はE2エミュレータ/IE850Aエミュレータのみサポートする。E2エミュレータ/IE850Aエミュレータサポート時には、2.2.1の起動手順に従って呼び出すこと。

(2) mode = EX\_EXTPWR\_USERIF\_ASPIFはターゲットシステムにもASP I/Fにも電源供給するが、E2エミュレータが供給可能な電流は200mAまでである。RHのターゲットシステムだけでも電流消費が大きいため、EX\_EXTPWR\_USERIF\_ASPIFの使用は推奨しない(RLとの互換のため本modeは呼び出し可能としておく)。

(3) IE850Aの電源供給エラー(EX\_FE\_VOUTERROR, EX\_FE\_BEFOREVOUTERROR)に関して、デバッグ開始時にエラーが発生した場合はデバッグ開始できないようにすること。デバッグ中にエラーが発生した場合はエラーを返すのみ行うこと。デバッグ中にエラーが発生した場合にデバッガが自動でデバッグを終了し切断しないこと。

### ハードウェア条件設定

ハードウェア条件設定関数には、状態検出用と環境設定用の関数がある。

IEの状態を監視するための関数

ex\_getenv, ex\_rh\_getenv, ex\_getstatus, ex\_setcallback\_sts, ex\_boardinfo, ex\_boardinfo2

IEの環境設定をするための関数

ex\_setenv, ex\_initimem, ex\_clock, ex\_mapping, ex\_pincont, ex\_m\_montimeout,

ex\_socunitinfo, ex\_socunitinfo2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_getenv | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

Get environmental information

[Format]

int ex\_getenv( EXENV \*env )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | Attributes | Description | 対象 |
| EXENV | \*env | < O> | 環境情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | char | pack\_info[48] | <--> | -------------- | -- |
|  | char | miro\_info[48] | <--> | -------------- | -- |
|  | char | exec\_info[48] | < O> | EXECバージョン情報 | ○ |
|  | char | moni\_info[48] | <--> | -------------- | -- |
|  | char | conf\_info[48] | <--> | -------------- | -- |
|  | char | flash\_info[48] | <--> | -------------- | -- |
|  | ULONG | trcmem\_size | <--> | -------------- | -- |
|  | USHORT | btrcmem\_size | <--> | -------------- | -- |
|  | USHORT | prt\_num | <--> | -------------- | -- |
|  | ULONG | delay\_num | <--> | -------------- | -- |
|  | USHORT | rprm\_size | <--> | -------------- | -- |
|  | USHORT | seq\_num | <--> | -------------- | -- |
|  | USHORT | pass\_num | <--> | -------------- | -- |
|  | USHORT | snap\_num | <--> | -------------- | -- |
|  | USHORT | stub\_num | <--> | -------------- | -- |
|  | USHORT | swdmm\_num | <--> | -------------- | -- |
|  | USHORT | hwdmm\_num | <--> | -------------- | -- |
|  | UCHAR | evn\_num | <--> | -------------- | -- |
|  | UCHAR | busevn\_num | <--> | -------------- | -- |
|  | UCHAR | fetchevn\_num | <--> | -------------- | -- |
|  | UCHAR | brdevn\_num | <--> | -------------- | -- |
|  | UCHAR | timer\_num | <--> | -------------- | -- |
|  | UCHAR | brk\_mode | <--> | -------------- | -- |
|  | UCHAR | em1\_chk | <--> | -------------- | -- |
|  | UCHAR | exec\_chk | <--> | -------------- | -- |
|  | USHORT | coreboard\_chk | <--> | -------------- | -- |
|  | ULONG | extsupport\_chk | <--> | -------------- | -- |
|  | USHORT | prom\_chk | <--> | -------------- | -- |
|  | USHORT | ieid | <--> | -------------- | -- |
|  | UCHAR | trcbusevn\_num | <--> | -------------- | -- |
|  | ULONG | dmrom\_addr | <--> | -------------- | -- |
|  | ULONG | dmrom\_size | <--> | -------------- | -- |
| }EXENV | | |  |  |  |

[Arguments]

|  |  |
| --- | --- |
| exec\_info[48] | |
|  | EXECの品種、製品名、バージョンの文字列 |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

[Description]

exec\_info[]は、"RH850G3 Executor Vx.xxz [dd mmm yyyy]"の文字列を返す。xは実際のバージョン、zはリビジョン、ddは日、mmmは月のアルファベット3文字略号、yyyyは西暦が入る。バージョンV1.20より、バージョン表記が変更され、常にVバージョンとなった。

E1系にあった"Copyright 2008"という文字列は削除する。ID850QBでは使われないのと、サードパーティ製デバッガではこの文字列をそのまま表示しているものもあり、社名記載がないので意味をなしていないため。また、社名まで入れると文字数をオーバーしてしまうため削除とする。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_getenv | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

Get environmental information

[Format]

int ex\_rh\_getenv(EXGETENV\_MCU \*envmcu, EXGETENV\_FUNC \*envfunc, EXGETENV\_EMU \*envemu)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXGETENV\_MCU | \*envmcu | < O> | DCU module mounting information in MCU | ○ |
| EXGETENV\_FUNC | \*envfunc | < O> | Debug function support information | ○ |
| EXGETENV\_EMU | \*envemu | < O> | Emulator general information | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mcuinfo\_num | < O> | MCU内デバッグモジュールの搭載情報定義数 | ○ |
|  | ULONG | mcuinfo[32] | < O> | MCU内デバッグモジュールの搭載情報 | ○ |
| }EXGETENV\_MCU | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | funcinfo\_num | < O> | デバッグ機能の制限情報定義数 | ○ |
|  | ULONG | funcinfo[32] | < O> | デバッグ機能の制限情報 | ○ |
| }EXGETENV\_FUNC | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | emuinfo\_num | < O> | エミュレータ全般情報の定義数 | ○ |
|  | ULONG | emuinfo[32] | < O> | エミュレータ全般の情報 | ○ |
| }EXGETENV\_EMU | | |  |  |  |

[Arguments]

|  |  |
| --- | --- |
| mcuinfo\_num | |
|  | MCU内デバッグモジュールの搭載情報定義総数。mcuinfo[]の有効要素数を定義する。  Total number of mounted information definitions for debug modules in the MCU. Define the number of valid elements of mcuinfo []. |
|  | |
| mcuinfo[32] | |
|  | MCU内デバッグモジュールの搭載情報  Mount information of debug module in MCU |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mcuinfo[0] | 内蔵トレースRAMのサイズ(バイト単位)  Internal trace RAM size (in bytes) | | |  |  |
|  |  |  | | |  |  |
|  | mcuinfo[1] | 内蔵トレースのモジュール搭載情報  Module mounting information of internal trace | | | |  |
|  |  | 【RH850 G3世代MCUの場合】 | | | |  |
|  |  | b0  DBG\_LUT(b4) | | CPUコアPE1用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b1  DBG\_LUT(b4) | | CPUコアPE2用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b2  DBG\_LUT(b4) | | CPUコアPE3用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b3  DBG\_LUT(b4) | | CPUコアPE4用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b4  DBG\_LUT(b4) | | CPUコアPE5用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b5  DBG\_LUT(b4) | | CPUコアPE6用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b6  DBG\_LUT(b18) | | トレースRAM(TCU\_TRM)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b7  ~~DBG\_LUT(b8)~~ | | 未使用 (0固定)  ~~ソフトトレース(TCU\_SFT)出力の実装状況~~  ~~0: 搭載していない~~  ~~1: 搭載している~~ | |  |
|  |  | b8  DBG\_LUT(b17) | | トレース出力IF(TCU\_TRO)の実装状況  0: 搭載していない  1: 搭載している | |  |
|  |  | b9-b31 | | 未使用(0固定) | |  |
|  |  |  | |  | |
|  |  | 【RH850 G4世代MCUの場合】 | | | |
|  |  | b0  DBG\_LUT(b4) | CPUコアPE0用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b1  DBG\_LUT(b4) | CPUコアPE1用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b2  DBG\_LUT(b4) | CPUコアPE2用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b3  DBG\_LUT(b4) | CPUコアPE3用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b4  DBG\_LUT(b4) | CPUコアPE4用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b5  DBG\_LUT(b4) | CPUコアPE5用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b6  DBG\_LUT(b4) | CPUコアPE6用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b7  DBG\_LUT(b4) | CPUコアPE7用トレースメッセージ生成部(TCU\_CPU)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | B8-b15 | 未使用(0固定) | | |
|  |  | b16  DBG\_LUT(b18) | トレースRAM(TCU\_TRM)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b17  DBG\_LUT(b19) | トレース出力IF(TCU\_ALO)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b18  DBG\_LUT(b17) | トレース出力IF(TCU\_TRO)の実装状況  0: 搭載していない  1: 搭載している | | |
|  |  | b19-b31 | 未使用(0固定) | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mcuinfo[2] | 内蔵トレースのスレーブ系モジュール搭載情報  Slave module mounting information of internal trace | |  |
|  |  | 【RH850 G3世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b23) | スレーブEMS用トレースメッセージ生成部(TCU\_EMS)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b6) | スレーブTCM用トレースメッセージ生成部(TCU\_TCM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b12) | スレーブGRM用トレースメッセージ生成部(TCU\_GRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b14) | スレーブGIO用トレースメッセージ生成部(TCU\_GIO)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b25) | スレーブGVC/KVC/AXI用トレースメッセージ生成部(TCU\_GVC/TCU\_KVC/TCU\_AXI)の実装状況  0: 搭載していない  1: 搭載している  ※TCU\_GVC, TCU\_KVC, TCU\_AXIは排他実装となる。 |  |
|  |  | b5-b31 | 未使用(0固定) |  |
|  |  |  |  |  |
|  |  | 【RH850 G4世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b6) | CPUコアPE0ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b6) | CPUコアPE1ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b6) | CPUコアPE2ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b6) | CPUコアPE3ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b6) | CPUコアPE4ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b6) | CPUコアPE5ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b6) | CPUコアPE6ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b6) | CPUコアPE7ローカルRAM用トレースメッセージ生成部(TCU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8  DBG\_LUT(b12) | クラスタCL0クラスタRAM用トレースメッセージ生成部(TCU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b9  DBG\_LUT(b12) | クラスタCL1クラスタRAM用トレースメッセージ生成部(TCU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b10  DBG\_LUT(b12) | クラスタCL2クラスタRAM用トレースメッセージ生成部(TCU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b11  DBG\_LUT(b12) | クラスタCL3クラスタRAM用トレースメッセージ生成部(TCU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b12  DBG\_LUT(b25) | 非CPUマスタ(DMA)用トレースメッセージ生成部(TCU\_DMA)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b13  DBG\_LUT(b28) | Trace Filter RAM制御モジュール(TRCFRAMC)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b14-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mcuinfo[3] | イベントのモジュール搭載情報  Event module mounting information | |  |
|  |  | 【RH850 G3世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b3) | CPUコアPE1用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b3) | CPUコアPE2用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b3) | CPUコアPE3用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b3) | CPUコアPE4用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b3) | CPUコアPE5用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b3) | CPUコアPE6用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6-b31 | 未使用(0固定) |  |
|  |  |  |  |  |
|  |  | 【RH850 G4世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b3) | CPUコアPE0用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b3) | CPUコアPE1用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b3) | CPUコアPE2用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b3) | CPUコアPE3用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b3) | CPUコアPE4用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b3) | CPUコアPE5用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b3) | CPUコアPE6用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b3) | CPUコアPE7用イベント(TEU\_CPU)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mcuinfo[4] | イベントのスレーブ系モジュール搭載情報  Event slave module mounting information | |  |
|  |  | 【RH850 G3世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b22) | スレーブEMS用イベント生成部(TEU\_EMS)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b5) | スレーブTCM用イベント生成部(TEU\_TCM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b11) | スレーブGRM用イベント生成部(TEU\_GRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b13) | スレーブGIO用イベント生成部(TEU\_GIO)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b24) | スレーブGVC/KVC/AXI用イベント生成部(TEU\_GVC/TEU\_KVC/TEU\_AXI)の実装状況  0: TEU\_GVC/TEU\_KVC/TEU\_AXIを搭載していない  1: TEU\_GVC/TEU\_KVC/TEU\_AXIを搭載している  ※TEU\_GVC, TEU\_KVC, TEU\_AXIは排他実装となる。 |  |
|  |  | b5-b31 | 未使用(0固定) |  |
|  |  |  |  |  |
|  |  | 【RH850 G4世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b5) | CPUコアPE0ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b5) | CPUコアPE1ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b5) | CPUコアPE2ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b5) | CPUコアPE3ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b5) | CPUコアPE4ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b5) | CPUコアPE5ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b5) | CPUコアPE6ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b5) | CPUコアPE7ローカルRAM用イベント生成部(TEU\_LRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8  DBG\_LUT(b11) | クラスタCL0クラスタRAM用イベント生成部(TEU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b9  DBG\_LUT(b11) | クラスタCL1クラスタRAM用イベント生成部(TEU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b10  DBG\_LUT(b11) | クラスタCL2クラスタRAM用イベント生成部(TEU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b11  DBG\_LUT(b11) | クラスタCL3クラスタRAM用イベント生成部(TEU\_CRM)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b12  DBG\_LUT(b24) | 非CPUマスタ(DMA)用イベント生成部(TEU\_DMA)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b13-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mcuinfo[5] | タイマのモジュール搭載情報  Timer module mounting information | |  |
|  |  | 【RH850 G3世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b0) | CPUコアPE1用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b0) | CPUコアPE2用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b0) | CPUコアPE3用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b0) | CPUコアPE4用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b0) | CPUコアPE5用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b0) | CPUコアPE6用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b2) | CPUコアPE1用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b2) | CPUコアPE2用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8  DBG\_LUT(b2) | CPUコアPE3用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b9  DBG\_LUT(b2) | CPUコアPE4用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b10  DBG\_LUT(b2) | CPUコアPE5用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b11  DBG\_LUT(b2) | CPUコアPE6用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b12-b31 | 未使用(0固定) |  |
|  |  |  |  |  |
|  |  | 【RH850 G4世代MCUの場合】 | |  |
|  |  | b0  DBG\_LUT(b0) | CPUコアPE0用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b1  DBG\_LUT(b0) | CPUコアPE1用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b0) | CPUコアPE2用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b0) | CPUコアPE3用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b0) | CPUコアPE4用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b0) | CPUコアPE5用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b0) | CPUコアPE6用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b0) | CPUコアPE7用TCKカウント タイマ部(TMU\_TCK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8-b15 | 未使用(0固定) |  |
|  |  | b16  DBG\_LUT(b2) | CPUコアPE0用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b17  DBG\_LUT(b2) | CPUコアPE1用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b18  DBG\_LUT(b2) | CPUコアPE2用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b19  DBG\_LUT(b2) | CPUコアPE3用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b20  DBG\_LUT(b2) | CPUコアPE4用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b21  DBG\_LUT(b2) | CPUコアPE5用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b22  DBG\_LUT(b2) | CPUコアPE6用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b23  DBG\_LUT(b2) | CPUコアPE7用CPUカウント タイマ部(TMU\_CLK)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b24-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mcuinfo[6] | ソフトトレースのモジュール搭載情報  Module information of soft trace | |  |
|  |  | b0  DBG\_LUT(b8) | CPUコアPE0用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している  Implementation status of soft trace control unit (TCU\_SFT) for CPU core PE0  0: Not installed  1: equipped |  |
|  |  | b1  DBG\_LUT(b8) | CPUコアPE1用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b2  DBG\_LUT(b8) | CPUコアPE2用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b3  DBG\_LUT(b8) | CPUコアPE3用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b4  DBG\_LUT(b8) | CPUコアPE4用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b5  DBG\_LUT(b8) | CPUコアPE5用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b6  DBG\_LUT(b8) | CPUコアPE6用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b7  DBG\_LUT(b8) | CPUコアPE7用ソフトトレース制御部(TCU\_SFT)の実装状況  0: 搭載していない  1: 搭載している |  |
|  |  | b8-b31 | 未使用(0固定) |  |

|  |  |
| --- | --- |
| funcinfo\_num | |
|  | デバッグ機能の制限情報定義数  Number of defined limit information of debug function |
| funcinfo[32] | |
|  | デバッグ機能の制限情報 |
|  | ここで定義する情報は、EXECを制御する上位モジュールの制御に影響する機能制限のみを  対象とする。  Debug function restriction information The information defined here is only the function restrictions that affect the control of the higher-level module that controls EXEC. set to target. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[0] | トレース機能のサポート情報  Trace function support information | |  |
|  |  | b0 | トレース タイムスタンプ機能のサポート状況  0: サポートしていない  1: サポートしている  Support status of trace time stamp function 0: not supported 1: Supported |  |
|  |  | b1 | JARL( jarl [reg1], reg3 )命令を実行した場合にトレースメッセージに分岐先PC情報が正しく出力されない不具合の状況  0: 不具合が該当する  1: 不具合が該当しない  ※ホットプラグインRUN中は不定値となる。  When the JARL (jarl [reg1], reg3) instruction is executed, the information of the branch destination PC is not output correctly in the trace message. 0: Defect applicable 1: Not applicable \* The value is undefined while the hot plug-in is running. |  |
|  |  | b2 | トレースメッセージ変換機能の実装の状況  0: 実装なし  1: 実装あり  Implementation status of trace message conversion function 0: No implementation 1: implemented |  |
|  |  | b3 | <TCU\_GRM/TCU\_CRM>と<TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>の排他利用に関する制限の状況  0: 制限なし  1: 制限あり  Status of restrictions on exclusive use of <TCU\_GRM / TCU\_CRM> and <TCU\_GVC / TCU\_KVC / TCU\_AXI / TCU\_DMA> 0: No limit 1: Limited |  |
|  |  | b4 | 未使用(0固定)  Unused (fixed at 0) |  |
|  |  | b5 | トレースメッセージに記録されるPEIDのアサイン設定の状況  0: 不要  1: 必要  Status of PEID assignment settings recorded in trace messages 0: Not required 1: necessary |  |
|  |  | b6 | PODのトレース用FIFOの書き込みカウント機能のサポート状況  0: サポートしていない  1: サポートしている  Support status of write count function of POD trace FIFO 0: not supported 1: Supported |  |
|  |  | b7 | グローバルトレースバッファ連続アクセス検出機能のサポート状況  0: サポートしていない  1: サポートしている  Support status of global trace buffer continuous access detection function 0: not supported 1: Supported |  |
|  |  | b8 | グローバルタイムスタンプ(絶対時間)計測カウンタの取得のサポート状況  0: サポートしていない  1: サポートしている  Support status of acquisition of global timestamp (absolute time) measurement counter 0: not supported 1: Supported |  |
|  |  | b9 | SPIDフィルタリング機能のサポート状況  0: サポートしていない  1: サポートしている  Support status of SPID filtering function   0: not supported   1: Supported |  |
|  |  | b10-b31 | 未使用(0固定) |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[1] | ブレーク機能のサポート情報  Break function support information | |  |
|  |  | b0 | ERAM(旧TRAM)資源を使用したソフトウェアブレークのサポート状況  0: サポートしていない  1: サポートしている |  |
|  |  | b1-b31 | 未使用(0固定) |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[2] | イベント機能のサポート情報  Event function support information | |  |
|  |  | b0 | 予約(0固定) |  |
|  |  | b1 | TEU\_CPUの命令系範囲イベントのサポート状況  0：サポートしていない  1：サポートしている  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support status of instruction range event of TEU\_CPU 0: Not supported 1: Supported \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) |  |
|  |  | b2 | CPUコアのデータ値を指定したアクセスイベントのサポート状況  0：サポートしていない  1：サポートしている  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support status of access events that specify CPU core data values 0: Not supported 1: Supported \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) |  |
|  |  | b3-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[3] | CPUトレースのデータトレースメッセージの出力タイプの情報  Output type information of CPU trace data trace message | |  |
|  |  | b0-b2  b3-b5  b6-b8  b9-b11  b12-b14  b15-b17  b18-b20  b21-b23 | PE0の出力タイプPE0 output type  PE1の出力タイプ  PE2の出力タイプ  PE3の出力タイプ  PE4の出力タイプ  PE5の出力タイプ  PE6の出力タイプ  PE7の出力タイプ  　000：G3M, G3K, G3K-Hタイプ  　001：G3M-Hタイプ  　010：G4MHタイプ  　011～111：予約 |  |
|  |
|  |  | b24-b31 | 予約(0固定) |  |
|  |  |  | * 出力タイプの詳細は、[引数説明]の"CPUトレースのデータトレースメッセージの出力タイプの詳細"の表を参照   \* For details of the output type, refer to the table of "Details of output type of CPU trace data trace message" in [Parameter description]. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[4] | 実行機能のサポート情報  Execution function support information | |  |
|  |  | b0 | 初期停止機能に関する情報  0：初期停止機能なし(ex\_resetrun使用不可)  1：初期停止機能あり(ex\_resetrun使用可)  Information about the initial stop function 0: No initial stop function (ex\_resetrun cannot be used) 1: With initial stop function (ex\_resetrun can be used) |  |
|  |  | b1-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[5] | 端子のマスク機能のサポート情報  Support information for pin mask function | |  |
|  |  | b0 | 外部リセットのマスク機能に関する情報  0：外部リセットのマスク機能なし  1：外部リセットのマスク機能あり |  |
|  |  | b1 | PWRGD信号のマスク機能に関する情報  0：PWRGD信号のマスク機能なし  1：PWRGD信号のマスク機能あり |  |
|  |  | b2-b31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[6] | 外部ソフトトレース単線出力機能のサポート情報(デバイス単位)  Support information of external soft trace single line output function (device unit) | |  |
|  |  | b0 | 外部ソフトトレース単線出力機能に関する情報  0：外部ソフトトレース単線出力は使用不可  1：外部ソフトトレース単線出力は使用可  Information on external soft trace single line output function 0: External soft trace single line output cannot be used 1: External soft trace single line output is available |  |
|  |  | b1 | 外部ソフトトレース単線出力のSPIDフィルタリング機能に関する情報  0: サポートしていない  1: サポートしている  Information on SPID filtering function of external soft trace single line output   0: not supported 1: Supported |  |
|  |  | b2-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[7] | デバッグ情報出力機能のサポート情報  Support information for the debug information output function | |  |
|  |  | b0 | デバッグ情報出力機能の使用可否に関する情報  0：デバッグ情報出力機能を使用不可  1：デバッグ情報出力機能を使用可  Information about whether the debug information output function can be used 0: Debug information output function cannot be used 1: Debug information output function can be used |  |
|  |  | b1-b31 | 未使用(0固定) |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[8] | 外部ソフトトレース単線出力機能のサポート情報(コア単位)１  Support information of external soft trace single line output function (core unit) 1  Information on disabling output of DBCP and DBTAG instructions | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | DBCP命令、DBTAG命令の出力無効化に関する情報  PE0：取得情報の対応Correspondence of acquired information  PE1：取得情報の対応  PE2：取得情報の対応  PE3：取得情報の対応  PE4：取得情報の対応  PE5：取得情報の対応  PE6：取得情報の対応  PE7：取得情報の対応  0：NMで実行されるDBCP命令、DBTAG命令の出力の無効設定に非対応  1：NMで実行されるDBCP命令、DBTAG命令の出力の無効設定に対応  0: Does not support invalidation of output of DBCP and DBTAG instructions executed by NM 1: Supports the invalid setting of the output of DBCP and DBTAG instructions executed by NM |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[9] | 外部ソフトトレース単線出力機能のサポート情報(コア単位)2  External soft trace single line output function support information (per core) 2 - Information on non-real-time trace mode support | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | ノンリアルタイムトレースモード対応に関する情報  PE0：取得種別の対応  PE1：取得種別の対応  PE2：取得種別の対応  PE3：取得種別の対応  PE4：取得種別の対応  PE5：取得種別の対応  PE6：取得種別の対応  PE7：取得種別の対応  0：ノンリアルタイムトレースモードに非対応  1：ノンリアルタイムトレースモードに対応  0: Does not support non-real-time trace mode 1: Supports non-real-time trace mode |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[10] | 外部ソフトトレース単線出力機能のサポート情報(コア単位)3  External soft trace single line output function support information (per core) 3 - Information on automatic start / stop response by event detection | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | イベント検出による自動開始／停止対応に関する情報  PE0：取得条件の対応  PE1：取得条件の対応  PE2：取得条件の対応  PE3：取得条件の対応  PE4：取得条件の対応  PE5：取得条件の対応  PE6：取得条件の対応  PE7：取得条件の対応  0：イベント検出による自動開始／停止に非対応  1：イベント検出による自動開始／停止に対応  ※ 本開発バージョンでは非対応(0)固定  0: Does not support automatic start / stop by event detection 1: Supports automatic start / stop by event detection \* Not supported (0) in this development version |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[11] | 外部ソフトトレース単線出力機能のサポート情報(コア単位)4  External soft trace single line output function support information (per core) 4 - Information on tag filtering support for group filtering | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | タグ値フィルタリング機能のグループフィルタリング対応に関する情報  PE0：グループフィルタリングの対応  PE1：グループフィルタリングの対応  PE2：グループフィルタリングの対応  PE3：グループフィルタリングの対応  PE4：グループフィルタリングの対応  PE5：グループフィルタリングの対応  PE6：グループフィルタリングの対応  PE7：グループフィルタリングの対応  0：グループフィルタリングに非対応  1：グループフィルタリングに対応  ※ 本開発バージョンでは非対応(0)固定 |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[12] | 時間計測機能(RUN-BRK)のサポート情報(コア単位)  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support information of time measurement function (RUN-BRK) (per core) \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | 時間計測機能(RUN-BRK)のコア単位での使用可否に関する情報  PE0の時間計測機能(RUN-BRK)の対応情報  PE1の時間計測機能(RUN-BRK)の対応情報  PE2の時間計測機能(RUN-BRK)の対応情報  PE3の時間計測機能(RUN-BRK)の対応情報  PE4の時間計測機能(RUN-BRK)の対応情報  PE5の時間計測機能(RUN-BRK)の対応情報  PE6の時間計測機能(RUN-BRK)の対応情報  PE7の時間計測機能(RUN-BRK)の対応情報  0：時間計測機能(RUN-BRK)に非対応  1：時間計測機能(RUN-BRK)に対応 |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[13] | 区間計測機能(DCUクロック利用)のサポート情報(コア単位)  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Section measurement function (using DCU clock) support information (per core) \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | 区間計測機能(DCUクロック利用)のコア単位での使用可否に関する情報  PE0の区間計測機能(DCUクロック利用)の対応情報  PE1の区間計測機能(DCUクロック利用)の対応情報  PE2の区間計測機能(DCUクロック利用))の対応情報  PE3の区間計測機能(DCUクロック利用)の対応情報  PE4の区間計測機能(DCUクロック利用)の対応情報  PE5の区間計測機能(DCUクロック利用)の対応情報  PE6の区間計測機能(DCUクロック利用)の対応情報  PE7の区間計測機能(DCUクロック利用)の対応情報  0：区間計測機能(DCUクロック利用)に非対応  1：区間計測機能(DCUクロック利用)に対応 |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[14] | 区間計測機能(CPUクロック利用)のサポート情報(コア単位)  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support information (per core) for section measurement function (using CPU clock) \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | 区間計測機能(CPUクロック利用)のコア単位での使用可否に関する情報  PE0の区間計測機能(CPUクロック利用)の対応情報  PE1の区間計測機能(CPUクロック利用)の対応情報  PE2の区間計測機能(CPUクロック利用))の対応情報  PE3の区間計測機能(CPUクロック利用)の対応情報  PE4の区間計測機能(CPUクロック利用)の対応情報  PE5の区間計測機能(CPUクロック利用)の対応情報  PE6の区間計測機能(CPUクロック利用)の対応情報  PE7の区間計測機能(CPUクロック利用)の対応情報  0：区間計測機能(CPUクロック利用)に非対応  1：区間計測機能(CPUクロック利用)に対応 |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[15] | パフォーマンス計測機能のサポート情報(コア単位)  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support information of performance measurement function (per core) \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7 | パフォーマンス計測機能のコア単位での使用可否に関する情報  PE0のパフォーマンス計測機能の対応情報  PE1のパフォーマンス計測機能の対応情報  PE2のパフォーマンス計測機能の対応情報  PE3のパフォーマンス計測機能の対応情報  PE4のパフォーマンス計測機能の対応情報  PE5のパフォーマンス計測機能の対応情報  PE6のパフォーマンス計測機能の対応情報  PE7のパフォーマンス計測機能の対応情報  0：パフォーマンス計測機能に非対応  1：パフォーマンス計測機能に対応 |  |
|  |  | b8-31 | 未使用(0固定) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[16] | TCU\_CPUのウォッチポイントタイムスタンプメッセージ(WPTSM)の生成機能のサポート状況  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support status of TCU\_CPU watchpoint timestamp message (WPTSM) generation function \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7  b8-31 | TCU\_CPU@PE0のWPTSMの生成機能のサポート状況  TCU\_CPU@PE1のWPTSMの生成機能のサポート状況  TCU\_CPU@PE2のWPTSMの生成機能のサポート状況  TCU\_CPU@PE3のWPTSMの生成機能のサポート状況  TCU\_CPU@PE4のWPTSMの生成機能のサポート状況  TCU\_CPU@PE5のWPTSMの生成機能のサポート状況  TCU\_CPU@PE6のWPTSMの生成機能のサポート状況  TCU\_CPU@PE7のWPTSMの生成機能のサポート状況  未使用 (0固定)  0: サポートしていない  1: サポートしている |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[17] | TCU\_LRMのウォッチポイントタイムスタンプメッセージ(WPTSM)の生成機能のサポート状況  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support status of TCU\_LRM watchpoint timestamp message (WPTSM) generation function \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4  b5  b6  b7  b8-31 | TCU\_LRM@PE0のWPTSMの生成機能のサポート状況  TCU\_LRM@PE1のWPTSMの生成機能のサポート状況  TCU\_LRM@PE2のWPTSMの生成機能のサポート状況  TCU\_LRM@PE3のWPTSMの生成機能のサポート状況  TCU\_LRM@PE4のWPTSMの生成機能のサポート状況  TCU\_LRM@PE5のWPTSMの生成機能のサポート状況  TCU\_LRM@PE6のWPTSMの生成機能のサポート状況  TCU\_LRM@PE7のWPTSMの生成機能のサポート状況  未使用 (0固定)  0: サポートしていない  1: サポートしている |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | funcinfo[18] | TCU\_CRM, TCU\_DMAのウォッチポイントタイムスタンプメッセージ(WPTSM)の生成機能のサポート状況  ※本情報は、G4世代MCUのみサポート(G3世代MCUでは参照不可)  Support status of TCU\_CRM, TCU\_DMA watchpoint timestamp message (WPTSM) generation function \* This information is supported only for G4 generation MCUs (not available for G3 generation MCUs) | |  |
|  |  | b0  b1  b2  b3  b4-15  b16  b17-31 | TCU\_CRM@CL0のWPTSMの生成機能のサポート状況  TCU\_CRM@CL1のWPTSMの生成機能のサポート状況  TCU\_CRM@CL2のWPTSMの生成機能のサポート状況  TCU\_CRM@CL3のWPTSMの生成機能のサポート状況  未使用 (0固定)  TCU\_DMAのWPTSMの生成機能のサポート状況  未使用 (0固定)  0: サポートしていない  1: サポートしている |  |
|  |  |  |  |  |

|  |  |
| --- | --- |
| emuinfo\_num | |
|  | エミュレータ全般の情報定義数  Number of information definitions for emulators in general |
| emuinfo[32] | |
|  | エミュレータ全般の情報  Emulator general information |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 【RH850 G3世代MCUの場合】 | |  |
|  | emuinfo[0] | CPUコアPE1用RCUのモニタ領域(デバッグ専用領域)のベースアドレス  Base address of monitor area (debug area) of RCU for CPU core PE1 |  |
|  | emuinfo[1] | CPUコアPE2用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[2] | CPUコアPE3用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[3] | CPUコアPE4用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[4] | CPUコアPE5用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[5] | CPUコアPE6用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  |  |  |  |
|  | 【RH850 G4世代MCUの場合】 | |  |
|  | emuinfo[0] | CPUコアPE0用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[1] | CPUコアPE1用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[2] | CPUコアPE2用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[3] | CPUコアPE3用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[4] | CPUコアPE4用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[5] | CPUコアPE5用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[6] | CPUコアPE6用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |
|  | emuinfo[7] | CPUコアPE7用RCUのモニタ領域(デバッグ専用領域)のベースアドレス |  |

[Return Values]

EX\_NOERROR 正常終了Successful completion

EX\_FE\_COMM EXECを起動していないEXEC not started

[Description]

[引数説明]

Argument description

(1)mcuinfo[]

MCU内デバッグモジュールの搭載情報は、MCUごとにデバッグ機能が変わる可能性があるモジュールを対象に搭載状況を定義する。

The mounting information of the debug module in the MCU defines the mounting status for modules that may have different debugging functions for each MCU.

(2)funcinfo[]

デバッグ機能の制限情報は、EXECを制御する上位モジュールの制御に影響する機能制限を対象とし、MCUごとで異なるデバッグ機能の制限情報を返送する。

なお、トレース機能を実装していないデバイスの場合は、funcinfo[0], [3], [16]-[18]は不定値を返す。また、funcinfo[6].b0(外部ソフトトレース単線出力機能のサポート情報)が使用不可(0)の時、funcinfo[8]-[11]は不定値を返す。

The debug function restriction information is intended for function restrictions that affect the control of the higher-level module that controls the EXEC, and returns different debug function restriction information for each MCU.  
For devices that do not have a trace function, funcinfo [0], [3], [16]-[18] return undefined values. Also, when funcinfo [6] .b0 (support information of external soft trace single line output function) is disabled (0), funcinfo [8]-[11] returns an undefined value

例. 上位モジュールで暫定対応を行っていた対処が、恒久対応では不要となるケース

JARL(jarl [reg1], reg3)命令を実行した場合、トレースメッセージに分岐先PC情報が出力できないMCU制限が発生するデバイスが存在する。上位モジュールはEXEC I/Fの制御がfuncinfo[0]b1によって暫定対応とするか恒久対応とするか区別する。

Example: A case where the provisional response was performed in the upper module but became unnecessary in the permanent response  
When the JARL (jarl [reg1], reg3) instruction is executed, there are devices that have an MCU restriction that cannot output branch destination PC information in the trace message. The upper module distinguishes whether the EXEC I / F control is provisional or permanent based on funcinfo [0] b1.

(制御対象のEXEC I/F： ① ex\_rh\_trctune trc\_tunetype=EX\_TRC\_CPUSYNCMSG trc\_tunecond[2]=2h

② ex\_rh\_trctune trc\_tunetype=EX\_TRC\_CPUMSGCHG trc\_tunecond[0]=1h

③ ex\_rh\_trctune trc\_tunetype=EX\_TRC\_PODCONVMSG trc\_tunecond[0]=1h)

[暫定対応] 上位モジュールは、EXEC I/Fを介してトレースメッセージ形式を変更し、MCU不具合が該当しないようにする必要がある。(トレースメッセージ形式を変更したことにより、トレースメッセージ量が増える)

出力トレースメッセージ：上記①の設定では、直接分岐/間接分岐は常に直接分岐同期化/間接分岐同期化メッセージで出力される。

また、トレースメッセージ変換機能を実装したPODを接続している場合に、②＋③の設定を行う(①は行わない)と、②の設定で直接分岐により出力された間接分岐メッセージは、③の設定により再度直接分岐メッセージに変換されて出力される。(間接分岐により出力されるメッセージは間接分岐メッセージで出力される。

注：ex\_rh\_getenvは各PEで同じ情報を返すI/Fだが、ex\_rh\_trctuneは各PEで設定を行うI/Fであるため、実装されているすべてのPEに対する設定が必要である。

[Temporary support] The upper module must change the trace message format via the EXEC I / F so that MCU failures do not apply. (A change in the trace message format increases the amount of trace messages.)  
Output trace message: In the setting of ① above, direct branch / indirect branch is always output as direct branch synchronization / indirect branch synchronization message.  
Also, if a POD that implements the trace message conversion function is connected, and the settings of (1) and (3) are not performed (the (2) is not performed), the indirect branch message output by the direct branch in (2) will be changed according to the setting in (3). It is directly converted into a branch message again and output. (A message output by an indirect branch is output as an indirect branch message.  
Note: ex\_rh\_getenv is an I / F that returns the same information in each PE, but ex\_rh\_trctune is an I / F that performs settings in each PE, so settings must be made for all implemented PEs.

[恒久対応] MCU改定により不具合を改修。

上位モジュールは、暫定対応で行っていたトレースメッセージ形式を変更する処理を行わない。

出力トレースメッセージ：間接分岐JARL 命令は間接分岐(JMP命令)として扱われて、間接分岐トレース(同期化)メッセージでトレースされる。(B-TYPE3~0＝3H：間接分岐(JMP命令、間接分岐JARL 命令))

[Permanent] Fixed bug by MCU revision.  
The upper module does not perform the process of changing the trace message format that was performed provisionally.  
Output trace message: Indirect branch JARL instruction is treated as an indirect branch (JMP instruction) and traced with an indirect branch trace (synchronization) message. (B-TYPE3 ~ 0 = 3H: Indirect branch (JMP instruction, indirect branch JARL instruction))

例. 上位モジュールでの暫定対応と恒久対応の制御が異なるケース

トレースのタイムスタンプ機能がMCU不具合により、サポートできないデバイスが存在する。上位モジュールは、funcinfo[0]b0によってMCU不具合の有無を区別する。

(制御対象のEXEC I/F：ex\_rh\_trceventsel2 \*item-b22)

Example: Case where control of provisional correspondence and permanent correspondence in upper module is different  
Some devices cannot support the trace time stamp function due to an MCU bug. The upper module distinguishes the presence / absence of an MCU failure by using funcinfo [0] b0.  
(EXEC I / F to be controlled: ex\_rh\_trceventsel2 \* item-b22)

[暫定対応] トレースのタイプスタンプ機能をEXEC I/Fの初期値により無効としている。

また、上位モジュールはEXEC I/F制御によっても有効にできない。

[Tentative] Trace timestamp function is disabled by the default value of EXEC I / F.  
Also, the upper module cannot be enabled by EXEC I / F control.

[恒久対応] MCU改定により不具合を改修。

上位モジュールは、EXEC I/F制御によってタイムスタンプ機能を有効にすることができる。

[Permanent] Fixed bug by MCU revision.  
The upper module can enable the time stamp function by EXEC I / F control

funcinfoの各情報に対して関連するEXEC I/F

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | ビット | 関連EXEC I/Fおよびパラメータ | 備考 |
| funcinfo[0] | b0 | ex\_rh\_trceventsel2 \*item b22 |  |
| b1 | ex\_rh\_trctune trc\_tunetype = EX\_TRC\_CPUSYNCMSG  ex\_socunitinfo2 unit\_type = EX\_UNIT\_TRACEOPT |  |
| b2 | ex\_rh\_trctune  trc\_tunetype = EX\_TRC\_CPUMSGCHG  trc\_tunetype = EX\_TRC\_PODCONVMSG  ex\_socunitinfo2 unit\_type = EX\_UNIT\_TRACEOPT |  |
| b3 | ex\_rh\_trceventsel2 \*item b23 |  |
| b4 | － |  |
| b5 | ex\_socunitinfo2 unit\_type = EX\_UNIT\_TRACEMSGPEID |  |
| b6 | ex\_rh\_trctune  trc\_tunetype = EX\_TRC\_PODFIFOCOUNT |  |
| b7 | ex\_rh\_trctune  trc\_tunetype = EX\_TRC\_GBUFCADET |  |
| b8 | ex\_rh\_trctune  trc\_tunetype = EX\_TRC\_GETGTS |  |
| b9 | ex\_rh\_trcfilter  filter\_type = EX\_FILTERTYPE\_SPID |  |
| funcinfo[1] | b0 | － |  |
| funcinfo[2] | b1 | ex\_rh\_evncond stat= EX\_M1\_STS |  |
| b2 | ex\_rh\_evncond stat= EX\_RUNB\_RW\_STS, EX\_RUNB\_R\_STS, EX\_RUNB\_W\_STS, EX\_RUNB\_RWND\_STS, EX\_RUNB\_RND\_STS, EX\_RUNB\_WND\_STS |  |
| funcinfo[3] | b0-b23 | ex\_rh\_trceventsel2 \*item b15:12 |  |
| funcinfo[4] | b0 | ex\_resetrun |  |
| funcinfo[5] | b0 | ex\_pincont mask\_fg[0] b14 |  |
| b1 | ex\_pincont mask\_fg[0] b15 |  |
| funcinfo[6] | b0 | ex\_socunitinfo2 unit\_type = EX\_UNIT\_TRCSEL |  |
| b1 | ex\_rh\_sfttrcfilter  filter\_type = EX\_FILTERTYPE\_SPID |  |
| funcinfo[7] | b0 | ex\_socunitinfo2 unit\_type = EX\_UNIT\_DBGINFOUT |  |
| funcinfo[8] | b6-1 | ex\_rh\_sfttrcmode \*cond b1:0 |  |
| funcinfo[9] | b6-1 | ex\_rh\_sfttrcmode \*cycle |  |
| funcinfo[10] | b6-1 | ex\_rh\_sfttrcmode \*mode, ex\_rh\_sfttrcevent | 本開発バージョンでは未サポート  Not supported in this development version |
| funcinfo[11] | b6-1 | ex\_rh\_sfttrcfilter filter\_type=EX\_FILTERTYPE\_TAG filter\_info[2] | 本開発バージョンでは未サポート |
| funcinfo[12] | b7-0 | ex\_timeread2 timer\_no=0 |  |
| funcinfo[13] | b7-0 | ex\_rh\_timeevent, ex\_rh\_timeread |  |
| funcinfo[14] | b7-0 | ex\_timeevent2, ex\_timeread2 timer\_no≠0 |  |
| funcinfo[15] | b7-0 | ex\_rh\_performanceset, ex\_rh\_performanceread, ex\_rh\_performanceclear, ex\_rh\_performancesw |  |
| funcinfo[16]  -[18] |  | ex\_rh\_trctune  trc\_tunetype = EX\_TRC\_WPTSMSG |  |

funcinfo[3]の"CPUトレースのデータトレースメッセージの出力タイプ"の情報によって、ex\_rh\_trceventsel2 \*itemb15:12に設定可能値が異なる。また、同一値を設定しても出力タイプが異なる場合がある。

注： (1)ex\_rh\_trceventsel2は各PEで設定が必要である。

(2)出力タイプは各PEで異なる場合があるので、ex\_rh\_trceventsel2 \*item15:12でも各PEで設定可能値が異なる場合がある。

CPUトレースのデータトレースメッセージの出力タイプの詳細

|  |  |  |
| --- | --- | --- |
| funcinfo[3] - b[N\*3]-b[N\*3+2]  (N=PE番号) | ex\_rh\_trceventsel2 \*item b15:12  の設定可能値 | 備考 |
| 0:G3M, G3K, G3K-Hタイプ | 0：EX\_RH\_EVN\_CPUDTM1\_1  1：EX\_RH\_EVN\_CPUDTM2\_2  2：EX\_RH\_EVN\_CPUDTM1\_35  3：EX\_RH\_EVN\_CPUDTM2\_45  4：EX\_RH\_EVN\_CPUDTM3\_3  5：EX\_RH\_EVN\_CPUDTM4\_4 |  |
| 1:G3M-Hタイプ | 0：EX\_RH\_EVN\_CPUDTM35\_35  1：EX\_RH\_EVN\_CPUDTM45\_45  4：EX\_RH\_EVN\_CPUDTM3\_3  5：EX\_RH\_EVN\_CPUDTM4\_4 | 2, 3を設定したときはエラー(EX\_PE\_DATA)を返す。 |
| 2:G4MHタイプ | 0：EX\_RH\_EVN\_CPUDTM1\_1  1：EX\_RH\_EVN\_CPUDTM2\_2  2：EX\_RH\_EVN\_CPUDTM35\_35  3：EX\_RH\_EVN\_CPUDTM45\_45  4：EX\_RH\_EVN\_CPUDTM3\_3  5：EX\_RH\_EVN\_CPUDTM4\_4 |  |

各出力タイプの詳細は、ex\_rh\_trceventsel2の[動作説明]を参照。

(3)emuinfo[]

エミュレータ全般(MCUを含む)の環境情報を返送する。

なお、トレース機能を実装していないデバイスの場合は、emuinfo[0]～[5]の値は不定を返す。

[動作説明]

MCUに内蔵しているデバッグモジュールの搭載情報や制限機能の情報を返す。

[マルチコア時の注意点]

(1)物理CPU(PE)の区別なく、同じ情報を返す。

[デバッガ開発者向け注意事項]

1. MCU内に搭載されるデバッグモジュールがMCU種別により、デバッグモジュールの実装が変動する可能性がある。本I/Fでは、「MCU内デバッグモジュールの搭載情報」パラメータより接続されているMCUのデバッグモジュールの搭載情報を返送する。

デバッグモジュールの搭載状況により、サポートできないデバッグ機能に関しては、ユーザ側の操作で機能選択できないようなユーザI/Fとなることが望ましい。

デバッグモジュールが非搭載のデバッグ機能に対して、EXEC I/Fを介してデバッグ機能を有効設定した場合は、エラーを返送する。

1. トレースのサポート機能は、mcuinfo[1]のパラメータ情報を組み合わせて下記の通り判断する。

[内蔵トレース機能]

①と②の条件が両方成立した場合、内蔵トレースがサポート可能と判断する。

【RH850 G3世代MCUの場合】

①トレース記録したいCPUコア用トレースメッセージ生成部(TCU\_CPU)の実装状況(mcuinfo[1] b0-b5)または、トレース記録したいスレーブ用トレースメッセージ生成部の実装状況(mcuinfo[2]の何れか)が実装されていること。

②トレースRAM(TCU\_TRM)の実装状況がセット(mcuinfo[1]b6=1)されていること。

【RH850 G4世代MCUの場合】

①トレース記録したいCPUコア用トレースメッセージ生成部(TCU\_CPU)の実装状況(mcuinfo[1] b0-b7)または、トレース記録したいスレーブ用トレースメッセージ生成部の実装状況(mcuinfo[2]の何れか)が実装されていること。

②トレースRAM(TCU\_TRM)の実装状況がセット(mcuinfo[1]b16=1)されていること。

[外部トレース機能]

①と②の条件が両方成立した場合、外部トレースがサポート可能と判断する。

【RH850 G3世代MCUの場合】

①トレース記録したいCPUコア用トレースメッセージ生成部(TCU\_CPU)の実装状況(mcuinfo[1] b0-b5)または、トレース記録したいスレーブ用トレースメッセージ生成部の実装状況(mcuinfo[2]の何れか)が実装されていること。

②トレース出力IF(TCU\_TRO)の実装状況がセット(mcuinfo[1]b8=1)されていること。

【RH850 G4世代MCUの場合】

①トレース記録したいCPUコア用トレースメッセージ生成部(TCU\_CPU)の実装状況(mcuinfo[1] b0-b7)または、トレース記録したいスレーブ用トレースメッセージ生成部の実装状況(mcuinfo[2]の何れか)が実装されていること。

②トレース出力IF(TCU\_ALO)の実装状況がセット(mcuinfo[1]b18=1)されていること。

1. 「ホットプラグインRUN中」以外の時に本I/Fを初めてコールするとき、ユーザプログラムが実行中であった場合は、ブレークをしてからプログラムを再実行する処理が行われる。
2. 「ホットプラグインRUN中」に本I/Fを呼び出した場合、funcinfo[0](トレースサポート情報)のb1(JARL( jarl [reg1], reg3 )命令を実行した場合にトレースメッセージに分岐先PC情報が正しく出力されない不具合の状況)は不定値となる。
3. funcinfo[4]のb0(初期停止機能有無情報)において、1(初期停止機能あり)が返る品種でのみex\_resetrunを呼び出すこと。0(初期停止機能なし)が返る品種でex\_resetrunを呼び出した場合の動作は保証しない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_getstatus | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

各種ステータス情報を読み出す

[Format]

int ex\_getstatus( EXSTAT \*stat )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXSTAT | \*stat | < O> | ステータス情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | cpustat | < O> | エミュレーションCPUステータス情報 | ○ |
|  | ULONG | exestat | < O> | エミュレーション機能に関するステータス情報 | ○ |
|  | ULONG | hwbrfact | < O> | 各ブレーク要因の発生状態情報 | ○ |
|  | ULONG | pc\_value | < O> | ブレーク時のプログラムカウンタ値 | ○ |
| }EXSTAT | | |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpustat:エミュレーションCPUステータス情報 | | | | |
|  | | | | |
|  | bit0 | cs\_wait | -------------- | ----------- |
|  | bit1 | cs\_holdrq | -------------- | ----------- |
|  | bit2 | -------------- | -------------- | ----------- |
|  | bit3 | cs\_stop | STOPモード(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit4 | cs\_reset | リセット状態(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit5 | cs\_halt | HALTモード | 0:OFF, 1:ON |
|  | bit6 | cs\_vm\_mode | 仮想マシンモード | 0:OFF, 1:ON |
|  | bit7 | cs\_brk\_vm\_mode | ブレーク時の仮想マシンモードからの遷移 | 0:OFF, 1:ON |
|  | bit8 | cs\_fetch\_stop | 初期停止状態 | 0:OFF, 1:ON |
|  | bit9 | cs\_deep\_stop | DEEP STOPモード(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit10 | cs\_cyclic\_run | Cyclic RUNモード(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit11 | cs\_cyclic\_stop | Cyclic STOPモード(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit12 | cs\_cyclic\_disable | Cyclic Disableモード(検出可否はデバイス依存) | 0:OFF, 1:ON |
|  | bit13-15 | cs\_vm\_gpid | ゲストモード時のゲスト・パーティション識別子 | 0～7 |
|  | bit16-31 | -------------- | -------------- | ----------- |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| exestat:エミュレーション機能に関するステータス情報 | | | | |
|  | | | | |
|  | bit0 | es\_usrpro\_run | ユーザプログラム実行状態 | 0:BREAK, 1:RUN |
|  | bit1 | -------------- | -------------- | ----------- |
|  | bit2 | es\_target\_off | ターゲット電源の状態 | 0:ON, 1:OFF |
|  | bit3 | es\_trc\_run | トレース起動中 | 0:停止, 1:実行 |
|  | bit4 | es\_trc\_wraparound | トレースメモリへのラップアラウンド発生状態  ※E1/E20/E2使用時  トレース起動中は0固定とする。トレース停止時に値を更新する。  ※IE850/IE850A使用時  トレース起動中にも更新する。 | 0:なし, 1:発生 |
|  | bit5-7 | -------------- | -------------- | ----------- |
|  | bit8 | es\_tckcount1\_run | TMU\_TCKカウンタ1 | 0:停止, 1:実行 |
|  | bit9 | es\_tckcount2\_run | TMU\_TCKカウンタ2 | 0:停止, 1:実行 |
|  | bit10 | es\_tckcount3\_run | TMU\_TCKカウンタ3 | 0:停止, 1:実行 |
|  | bit11-13 | -------------- | -------------- | ----------- |
|  | bit14 | es\_vm\_timeout | [G4世代で仮想化支援機能有効時のみ]  デバッグ対象外のCPU動作モードでブレークした場合にデバッグ対象のCPU動作モードまで遷移させる機能が有効な時(ex\_soucuintinfo2、EX\_UNIT\_HEGESELのbit31)、タイムアウトで遷移に失敗し、デバッグ対象外のCPU動作モードでブレークした。  デバッグ対象のCPU動作モードまでステップする機能が有効な時(ex\_soucuintinfo2、EX\_UNIT\_HEGESELのbit30)、タイムアウトで遷移に失敗し、デバッグ対象外のCPU動作モードでブレークした。 | 0:なし, 1:発生 |
|  | bit15-17 | -------------- | -------------- | ----------- |
|  | bit18 | es\_cvd\_run | 未サポート | ----------- |
|  | bit19 | -------------- | -------------- | ----------- |
|  | bit20 | es\_clkcount1\_run | TMU\_CLKカウンタ1 | 0:停止, 1:実行 |
|  | bit21 | es\_clkcount2\_run | TMU\_CLKカウンタ2 | 0:停止, 1:実行 |
|  | bit22 | es\_clkcount3\_run | TMU\_CLKカウンタ3 | 0:停止, 1:実行 |
|  | bit23 | es\_clkcount4\_run | TMU\_CLKカウンタ4 | 0:停止, 1:実行 |
|  | bit24 | -------------- | -------------- | ----------- |
|  | bit25 | es\_trc\_over | トレースメッセージロスト発生状態  ※E1/E20/E2使用時  検出できないため0固定とする。  ※IE850/IE850A使用時  トレース記録中に一度でもロストが発生すると  “ 1 ”となる。  [注意]  記録したトレースにロストメッセージが含まれていない場合でも、ロストメッセージが1度でも発行されれば、”1”(ロスト発生)となる。  [クリアタイミング]  ex\_trcclearの発行または、トレース機能の設定を変更するI/Fを発行した場合に本フラグをクリアする。 | 0:なし, 1:発生 |
|  | bit26 | es\_trc\_overfullbreak | トレースフルブレーク時にトレースメモリがオーバーフローしてトレースパケットが消失。 | 0:なし, 1:発生 |
|  | bit27-31 | -------------- | -------------- | ----------- |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hwbrfact:イベントブレーク要因 | | | | |
|  | | | | |
|  | bit0 | es\_pwoff\_bk | ターゲット電源OFFによるブレーク | 0:なし, 1:発生 |
|  | bit1 | es\_relay\_bk | リレーブレーク | 0:なし, 1:発生 |
|  | bit2 | es\_expfuncact\_brk | 拡張機能のアクションブレーク | 0:なし, 1:発生 |
|  | bit3-4 | -------------- | -------------- | ----------- |
|  | bit5 | es\_expfuncfull\_bk | 拡張機能のフルブレーク | 0:なし, 1:発生 |
|  | bit6-9 | -------------- | -------------- | ----------- |
|  | bit10 | es\_trcdly\_bk | トレースディレイブレーク | 0:なし, 1:発生 |
|  | bit11-14 | -------------- | -------------- | ----------- |
|  | bit15 | es\_trcfull\_bk | トレースフルブレーク | 0:なし, 1:発生 |
|  | bit16 | es\_sfttrcfull\_bk | 外部ソフトトレース単線出力フルブレーク | 0:なし, 1:発生 |
|  | bit17-19 | -------------- | -------------- | ----------- |
|  | bit20 | es\_step\_bk | ステップ実行ブレーク | 0:なし, 1:発生 |
|  | bit21 | es\_soft\_bk | ソフトウェアブレーク | 0:なし, 1:発生 |
|  | bit22 | -------------- | -------------- | ----------- |
|  | bit23 | es\_enf\_bk | 強制ブレーク | 0:なし, 1:発生 |
|  | bit24 | es\_brkevnt\_bk | イベントブレーク | 0:なし, 1:発生 |
|  | bit25 | es\_timtotovf\_bk | TMU\_CLKタイマ積算カウント閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit26 | es\_timmaxovf\_bk | TMU\_CLKタイマ最大カウント閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit27 | es\_timminudf\_bk | TMU\_CLKタイマ最小カウント閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit28 | es\_timcntovf\_bk | TMU\_CLKタイマパスカウント閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit29 | es\_timtcktvfl\_bk | TMU\_TCKタイマ閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit30 | ex\_pfmcnttvfl\_bk | パフォーマンスカウント閾値違反ブレーク | 0:なし, 1:発生 |
|  | bit31 | -------------- | -------------- | ----------- |
|  | | | | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_FE\_M\_MONTIMOUT\_REG レジスタ関数モニタタイムアウト

EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED 初期停止中のため同期ブレークできない

EX\_FE\_FANERROR IE850Aが冷却ファン停止を検出しました。

ターゲット及びIE850Aの電源をOFFしてください

EX\_FE\_VOUTERROR IE850AがVOUTからの電源供給で過電流を検出しました

EX\_FE\_BEFOREVOUTERROR IE850AがVOUTから電源供給を開始できませんでした

[Description]

[引数説明]

(1)cpustat

CPUのステータス情報をビット単位で返す。

bit3：STOPモードであるかを示す。

(0：STOPモードでない 1：STOPモード)

bit4：リセット状態であるかを示す。

(0：リセット状態でない 1：リセット状態)

bit5：HALTモードであるかを示す。

(0：HALTモードでない 1：HALTモード)

bit6：仮想マシンモード(G4ではCPU動作モード)の状態を示す。

RH850G3 (0：ネイティブマシンモード 1：仮想マシンモード)

RH850G4 (0：ホストモードor従来モード　1：ゲストモード)

bit7：ブレーク時の仮想マシンモードからの遷移の状態を示す。

(0：仮想マシンモードからの遷移ではない 1：仮想マシンモードからの遷移)

注：ユーザプログラム実行中は、0を示す。

仮想化支援機能が有効な場合に、bit6-7が有効になる。仮想化支援機能の有効無効はex\_socunitinfo2(EX\_UNIT\_VIRTUALINFO)にて指定する。

RH850G4では未使用

bit8：初期停止状態を示す。

(0：初期停止状態でない 1：初期停止状態)

bit9：DEEP STOPモードであるかを示す。

(0：DEEP STOPモードでない 1：DEEP STOPモード)

bit10：Cyclic RUNモードであるかを示す。

(0：Cyclic RUNモードでない 1：Cyclic RUNモード)

bit11：Cyclic STOPモードを示す。

(0：Cyclic STOPモードでない 1：Cyclic STOPモード)

bit12：Cyclic RUNモードまたはCyclic STOPモードかつコアが無効状態を示す。

(0：無効状態でない 1：無効状態)

bit13-15：ゲストモード時のゲスト・パーティション識別子(GPID)を示す。RH850G3では未使用。

(0～7)

(2)exestat

エミュレーション機能に関するステータス情報をビット単位で返す。

bit0 ：ユーザプログラムの実行状態を示す。

(0：ブレーク中 1：実行中)

bit2 ：ターゲット電源の状態を示す。

(0：電源ON 1：電源OFF)

※IE850使用時でターゲット未接続の場合は、電源OFFの状態に固定する。

bit3 ：トレースの実行状態を示す。

(0：停止中 1：実行中)

bit4 ：トレースメモリへのラップアラウンドの発生状態を示す。

(0：発生なし 1：発生)

※ 本フラグは、トレースRAMの最終アドレスに書き込んだときに1：発生となる。従って、最終アドレスを書き込んだと同時にトレースを停止した場合は、トレースメッセージの上書きが発生していなくても1：発生となる。

bit8 ：TMU\_TCKカウンタ1の状態を示す。

(0：停止中 1：実行中)

bit9 ：TMU\_TCKカウンタ2の状態を示す。

(0：停止中 1：実行中)

bit10：TMU\_TCKカウンタ3の状態を示す。

(0：停止中 1：実行中)

bit14：G4世代の仮想化支援機能のデバッグ機能で、ブレーク後にデバッグ対象のCPU動作モードへ遷移できなかった状態を示す。

(0：デバッグ対象のCPU動作モードに遷移できた、または、デバッグ対象のCPU動作モードへ遷移させる機能がOFF  
 1：デバッグ対象のCPU動作モードに遷移できなかった)

bit20：TMU\_CLKカウンタ1の状態を示す。

(0：停止中 1：実行中)

bit21：TMU\_CLKカウンタ2の状態を示す。

(0：停止中 1：実行中)

bit22：TMU\_CLKカウンタ3の状態を示す。

(0：停止中 1：実行中)

bit23：TMU\_CLKカウンタ4の状態を示す。

(0：停止中 1：実行中)

bit25：トレースメッセージの取りこぼしの発生状態を示す。

(0：取りこぼし発生なし 1：取りこぼし発生)

bit26：トレースフルブレーク時のブレーク後のトレースメッセージの取りこぼしの発生状態を示す。

(0：取りこぼし発生なし 1：取りこぼし発生)

上記以外のbit：未使用(0固定)

(3)hwbrfact

各ブレーク要因の発生状態をビット単位で返す。

なお、同時に発生した場合は優先順位の高い要因を格納する。優先順位については、表 4‑1参照。

0：該当するビットのブレーク要因の発生なし

1：該当するビットのブレーク要因が発生

bit0 ：ターゲット電源OFFによるブレーク発生状態を示す。

bit1 ：リレーブレーク要因によるブレーク発生状態を示す。

bit2 ：拡張機能のアクションによるブレーク発生状態を示す。

bit5 ：拡張機能のメモリフルによるブレーク発生状態を示す。

bit10：トレースディレイ要因によるブレーク発生状態を示す。

なお、内蔵トレース選択時は常に0を示す。

bit15：トレースフル要因によるブレーク発生状態を示す。

bit16：外部ソフトトレース単線出力フル要因によるブレーク発生状態を示す。

拡張機能が有効時は、本ビットは常に0を返す。このときはb5を参照する。

bit20：ステップ実行要因によるブレーク発生状態を示す。

bit21：ソフトウェアブレーク要因によるブレーク発生状態を示す。

bit23：強制ブレーク要因によるブレーク発生状態を示す。

bit24：ハードウェアブレーク要因/イベント要因によるブレーク発生状態を示す。

bit25：TMU\_CLKタイマの積算カウントの閾値違反ブレーク発生状態を示す。

bit26：TMU\_CLKタイマの最大カウントの閾値違反ブレーク発生状態を示す。

bit27：TMU\_CLKタイマの最小カウントの閾値違反ブレーク発生状態を示す。

bit28：TMU\_CLKタイマのパスカウントの閾値違反ブレーク発生状態を示す。

bit29：TMU\_TCKタイマの閾値違反ブレーク発生状態を示す。

bit30：パフォーマンス計測のカウント閾値違反ブレーク発生状態を示す。

上記以外のbit：未使用(0固定)

(4)pc\_value

ブレーク発生時のプログラムカウンタ値を返す。

[動作説明]

CPUステータス情報、エミュレーション機能に関するステータス情報、各ブレーク要因の発生状態情報およびブレーク時のプログラムカウンタ値を返す。

なお、ハードウェアブレーク要因/イベント要因のイベント番号などの詳細はex\_hwbrkevent関数で返す。

同期デバッグモードにおいて、初期停止中にデバイスがイベントブレークなどで自発的にブレークした後に呼び出された場合は、エラー(EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED)を返す。

IE850Aの場合に、IE850Aの冷却ファン停止が検出されている時、エラー(EX\_FE\_FANERROR)を返す。冷却ファン停止が1回でも検出された場合、その後に冷却ファンが再稼動してもエラー(EX\_FE\_FANERROR)を返し続ける。

IE850Aの場合に、IE850Aからの供給電圧が異常値となっていた場合、エラー(EX\_FE\_VOUTERROR)を返す。

IE850Aの電源をONにした時、VOUT0またはVOUT1に0.5V以上の電圧が検出された場合も、供給電圧が異常である判断してエラー(EX\_FE\_BEFOREVOUTERROR)を返す。

IE850Aの供給電圧が異常であることを検出し、VCCLEDが消灯している状態では、IE850Aは電源供給を停止し、電源供給機能設定の変更も受け付けない。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・ユーザの操作によっては、ユーザプログラム停止中でもCPUステータス情報(cpustat)が変化する可能性があるため、定期的なex\_getstatusコマンド発行または非同期通信インターフェースのコールバック関数を使用してチェックすることを推奨する。

・ユーザプログラム停止中でかつ初期停止・CyclicRunに伴うCyclicDisableのコアを選択している場合、pc\_valueにはダミー値(0x00000000)を返す。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)の情報を返す。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[デバッガ開発者向け注意事項]

(1)ターゲット電源が電源OFF状態(exestat.es\_target\_off=1)の場合、ユーザプログラム実行状態ビットには実行中(exestat.esusrpro\_run=1)、プログラムカウンタ値(pc\_value)には無効な値を返す。また、トレースの実行状態ビットは電源OFF状態になる前の状態を保持する。電源OFF前がトレース起動中だった場合は電源OFF中もトレース起動中(exestat.es\_trc\_run=1)を返す。(CS+がこのステータスを見てトレース取得しないようにするための対策)

(2)ユーザプログラム実行中(exestat.esusrpro\_run=1)の場合、プログラムカウンタ値(pc\_value)には無効な値を返す。

(3)トレース関連の情報は、現在選択されているトレース出力タイプ(外部トレース、内蔵トレース)の情報を返す。なお、トレース出力タイプの変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

(4)「初期停止中のため同期ブレークできない」(EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED)の場合は、継続してデバッグを進めることができないため、ex\_resetを入れるようにユーザにガイドする。

(5) IE850Aの電源供給エラー(EX\_FE\_VOUTERROR, EX\_FE\_BEFOREVOUTERROR)および冷却ファンエラー(EX\_FE\_FANERROR)に関して、デバッグ開始時にエラーが発生した場合はデバッグ開始できないようにすること。デバッグ中にエラーが発生した場合はエラーを返すのみ行うこと。デバッグ中にエラーが発生した場合にデバッガが自動でデバッグを終了し切断しないこと。

(6)拡張機能が有効時かつユーザプログラムが実行中(exestat:b0=1)の場合は、cpustat, exestat, hwbrfactの他のビットは参照してはならない。参照した場合は不定となる。表 3‑4参照。

(7)拡張機能が無効時かつ外部ソフトトレース単線出力が有効時で、ユーザプログラムが実行中(exestat:b0=1)の場合は、cpustat, exestat, hwbrfactの他のビットは参照してはならない。参照した場合は不定となる。表 3‑4参照。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

表 3‑4　E2拡張機能および外部ソフトトレース単線出力機能と各ステータス仕様

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **分類** | **bit** | **bit名** | **E2拡張機能** | **無効** | | | | **有効** | | **備考** |
| **外部ソフトトレース単線出力機能** | **無効** | | **有効** | | **－** | |
| **実行状態** | **停止中** | **実行中** | **停止中** | **実行中** | **停止中** | **実行中** |
| cpustat | 0 | cs\_wait | | ○ | ○ | ○ | × | ○ | × |  |
| 1 | cs\_holdrq | | ○ | ○ | ○ | × | ○ | × |  |
| 3 | cs\_stop | | ○ | ○ | ○ | × | ○ | × |  |
| 4 | cs\_reset | | ○ | ○ | ○ | × | ○ | × |  |
| 5 | cs\_halt | | ○ | ○ | ○ | × | ○ | × |  |
| 6 | cs\_vm\_mode | | ○ | ○ | ○ | × | ○ | × |  |
| 7 | cs\_brk\_vm\_mode | | ○ | － | ○ | × | ○ | × |  |
| 8 | cs\_fetch\_stop | | ○ | ○ | ○ | × | ○ | × |  |
| 9 | cs\_deep\_stop | | ○ | ○ | ○ | × | ○ | × |  |
| 10 | cs\_cyclic\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 11 | cs\_cyclic\_stop | | ○ | ○ | ○ | × | ○ | × |  |
| 12 | cs\_cyclic\_disable | | ○ | ○ | ○ | × | ○ | × |  |
| 13-15 | cs\_vm\_gpid | | ○ | ○ | ○ | × | ○ | × |  |
| exestat | 0 | es\_usrpro\_run | | ○ | ○ | ○ | ○ | ○ | ○ |  |
| 2 | es\_target\_off | | ○ | ○ | ○ | × | ○ | × |  |
| 3 | es\_trc\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 4 | es\_trc\_wraparound | | ○ | ○※1 | ○ | × | ○ | × |  |
| 8 | es\_tckcount1\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 9 | es\_tckcount2\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 10 | es\_tckcount3\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 14 | es\_vm\_timeout | | ○ | ○ | － | － | － | － |  |
| 18 | es\_cvd\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 20 | es\_clkcount1\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 21 | es\_clkcount2\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 22 | es\_clkcount3\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 23 | es\_clkcount4\_run | | ○ | ○ | ○ | × | ○ | × |  |
| 25 | es\_trc\_over | | ○※1 | ○※1 | － | × | － | × |  |
| 26 | es\_trc\_overfullbreak | | × | × | × | × | × | × |  |
| hwbrfact | 0 | es\_pwoff\_bk | | ○ | － | ○ | × | ○ | × |  |
| 1 | es\_relay\_bk | | ○ | － | ○ | × | ○ | × |  |
| 2 | es\_expfuncact\_brk | | × | × | × | × | ○ | × | E2拡張有効時専用 |
| 5 | es\_expfuncfull\_bk | | × | × | × | × | ○ | × | E2拡張有効時専用 |
| 10 | es\_trcdly\_bk | | ○ | － | ○ | × | ○ | × |  |
| 15 | es\_trcfull\_bk | | ○ | － | ○ | × | ○ | × |  |
| 16 | es\_sfttrcfull\_bk | | × | × | ○ | × | × | × | E2拡張無効かつソフトトレース単線出力機能有効時専用 |
| 20 | es\_step\_bk | | ○ | － | ○ | × | ○ | × |  |
| 21 | es\_soft\_bk | | ○ | － | ○ | × | ○ | × |  |
| 23 | es\_enf\_bk | | ○ | － | ○ | × | ○ | × |  |
| 24 | es\_brkevnt\_bk | | ○ | － | ○ | × | ○ | × |  |
| 25 | es\_timtotovf\_bk | | ○ | － | ○ | × | ○ | × |  |
| 26 | es\_timmaxovf\_bk | | ○ | － | ○ | × | ○ | × |  |
| 27 | es\_timminudf\_bk | | ○ | － | ○ | × | ○ | × |  |
| 28 | es\_timcntovf\_bk | | ○ | － | ○ | × | ○ | × |  |
| 29 | es\_timtcktvfl\_bk | | ○ | － | ○ | × | ○ | × |  |
| 30 | ex\_pfmcnttvfl\_bk | | ○ | － | ○ | × | ○ | × |  |
| pc\_value | | | | ○※3 | －※2 | ○※3 | × | ○※3 | × |  |

○：有効、×：無効(参照した場合は不定値を返す)、－：無効(参照した場合は0を返す)

※1：E1/E20/E2エミュレータ使用時は"－"となる。

※2：この場合は、0ではなく0xFFFFFFFFとなる。

※3：初期停止・スタンバイモードデバッグ環境が有効設定時、停止(初期停止・Cyclic Disable)中のコアはダミー値(0x00000000)となる。

[仮想マシン対応時の注意点(G3世代)]

1. cpustat.cs\_vm\_modeは、本関数呼び出し時のマシンモードを示す。
2. cpustat.cs\_brk\_vm\_modeは、本関数呼び出し時のブレーク時の仮想マシンモードからの遷移を示す。
3. cpustat.cs\_brk\_vm\_modeは、ユーザプログラム実行中は状態遷移を識別できないため、0を示す。
4. cpustat.cs\_vm\_modeおよびcpustat.cs\_brk\_vm\_modeは、仮想化支援機能が無効な場合は0を示す。

[非同期デバッグモード時の注意点]

(1)非同期デバッグモード時は、ユーザプログラム実行状態がPEごとに異なるため、各PEに切り替えて参照すること。

[拡張機能有効時の注意点]

拡張機能有効時のユーザプログラム実行状態において、CPUステータス情報(cpustat)は取得できないため、ステータス情報はすべて不定値を返す。

拡張機能有効時のユーザプログラム実行状態において、エミュレーション機能に関するステータス情報(exestat)のbit0(es\_usrpro\_run)以外の情報は取得できないため不定値を返す。詳細は、[デバッガ開発者向け注意事項]の表 3‑4を参照

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_setcallback\_sts | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

各種ステータス情報を返すコールバック関数のアドレス情報を設定/参照する。

[Format]

int ex\_setcallback\_sts( EXCALLBACK callback\_sts )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXCALLBACK | callback\_sts | <IO> | ステータス情報用コールバック関数のアドレス情報 | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| callback\_sts | | |
|  | コールバック関数のアドレス情報 | |
|  | NULL | コールバック関数を発行しない。(Default) |
|  | NULL以外 | 指定されたアドレス情報を元にコールバック関数を発行 |
|  |  | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

[Description]

[引数説明]

(1)callback\_sts

各種ステータス情報を返すコールバック関数のアドレス情報を指定する。なお、一度も指定されていない場合は、コールバック関数を発行しない。

[動作説明]

上位層のソフトウェアに対してステータス情報などを返すために非同期通信を行うコールバック関数のアドレス情報を設定する。

一度も本関数を発行していない、もしくは発行してもコールバック関数のアドレス情報がNULLである場合、ステータスが変化してもコールバック関数を発行しない。この場合、CPUステータスなど情報の取得にはex\_getstatus関数を使用する。

[デバッガ開発者向け注意事項]

(1)コールバック関数は、上位層のソフトウェアで実態を準備すること。なお、コールバック関数の型および動作仕様については、「非同期通信インターフェース仕様」を参照のこと。

(2)一度も本関数を発行していない、もしくは発行してもコールバック関数のアドレス情報がNULLである場合、ステータスが変化してもコールバック関数を発行しない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_boardinfo **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

ボードバージョンを得る

[Format]

int ex\_boardinfo( EXBOARDINFO \*info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXBOARDINFO | \*info | <--> | -------- | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | char | exec\_version[8] | <--> | -------- | -- |
|  | char | df\_chipname[28] | <--> | -------- | -- |
|  | char | df\_fileversion[8] | <--> | -------- | -- |
|  | ULONG | board\_info | <--> | -------- | -- |
|  | USHORT | product\_id | <--> | -------- | -- |
|  | USHORT | product\_ver | <--> | -------- | -- |
|  | USHORT | svrom\_ver | <--> | -------- | -- |
|  | USHORT | svrom\_extraver | <--> | -------- | -- |
|  | EXBOARDVER | board1 | <--> | -------- | -- |
|  | EXBOARDVER | board2 | <--> | -------- | -- |
|  | EXBOARDVER | board3 | <--> | -------- | -- |
|  | EXBOARDVER | board4 | <--> | -------- | -- |
|  | EXBOARDVER | board5 | <--> | -------- | -- |
|  | EXBOARDVER | board6 | <--> | -------- | -- |
|  | EXBOARDVER | board7 | <--> | -------- | -- |
|  | EXBOARDVER | board8 | <--> | -------- | -- |
|  | EXBOARDVER | board9 | <--> | -------- | -- |
|  | char | dummy[8] | <--> | -------- | -- |
| }EXBOARDINFO | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | Id | <--> | -------- | -- |
|  | USHORT | Ver | <--> | -------- | -- |
|  | USHORT | fpgaver | <--> | -------- | -- |
|  | USHORT | extraver | <--> | -------- | -- |
| }EXBOARDVER | | |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。ボードバージョン情報などの取得は、ex\_boardinfo2を使用すること。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_boardinfo2 | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

ボードバージョンを得る(拡張版)

[Format]

int ex\_boardinfo2( EXBOARDINFO2 \*info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXBOARDINFO2 | \*info | <IO> | ボード情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | char | exec\_infoname[28] | < O> | EXEC名称情報 | ○ |
|  | char | exec\_version[8] | < O> | EXECバージョン情報文字列 | ○ |
|  | char | df\_chipname[28] | < O> | デバイスファイル内のデバイス品種名/拡張情報の文字列 | ○ |
|  | char | df\_fileversion[8] | < O> | デバイスファイルのファイルバージョン情報文字列 | ○ |
|  | ULONG | board\_info | < O> | 接続エミュレータ情報 | ○ |
|  | USHORT | product\_id | < O> | エミュレータ製品ID情報 | ○ |
|  | USHORT | product\_ver | < O> | エミュレータ製品管理バージョン情報 | ○ |
|  | USHORT | svrom\_ver | < O> | コントロールボード用ファームウェアのバージョン情報 | ○ |
|  | USHORT | svrom\_extraver | < O> | コントロールボード用ファームウェアのバージョン情報 | ○ |
|  | EXBOARDVER | board1 | < O> | コートロールボード情報 | ○ |
|  | EXBOARDVER | board2 | < O> | 予備 | -- |
|  | EXBOARDVER | board3 | < O> | DBボードの情報 | -- |
|  | EXBOARDVER | board4 | < O> | スーパートレースボードの情報 | -- |
|  | EXBOARDVER | board5 | < O> | 予備 | -- |
|  | EXBOARDVER | board6 | < O> | 予備 | -- |
|  | EXBOARDVER | board7 | < O> | 予備 | -- |
|  | EXBOARDVER | board8 | < O> | 予備 | -- |
|  | EXBOARDVER | board9 | < O> | 予備 | -- |
|  | char | dummy[6] | < O> | 予備 | -- |
|  | UCHAR | target\_system | <I > | ターゲットシステムの種別 | ○ |
|  | UCHAR | trace\_lane | <I > | トレースのレーン数 | ○ |
|  | char | \*devfile\_inf | <I > | デバイスファイル名(フルパス) | ○ |
|  | EXCOMMPORT | port1 | < O> | 予備 | -- |
|  | EXCOMMPORT | port2 | < O> | 予備 | -- |
|  | EXCOMMPORT | port3 | < O> | 予備 | -- |
|  | EXCOMMPORT | port4 | < O> | 予備 | -- |
|  | EXCOMMPORT | port5 | < O> | 予備 | -- |
|  | EXCOMMPORT | port6 | < O> | 予備 | -- |
|  | EXCOMMPORT | port7 | < O> | 予備 | -- |
|  | EXCOMMPORT | port8 | < O> | 予備 | -- |
|  | EXCOMMPORT | port9 | < O> | 予備 | -- |
|  | char | dummy2[8] | < O> | 予備 | -- |
| }EXBOARDINFO2 | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | Id | < O> | ボードID | ○ |
|  | USHORT | Ver | < O> | ボードバージョン | ○ |
|  | USHORT | fpgaver | < O> | FPGAバージョン | ○ |
|  | USHORT | extraver | < O> | ボード予備バージョン | -- |
| }EXBOARDVER | | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | char | name[12] | < O> | 通信ポート名称 | -- |
|  | ULONG | support\_chk | < O> | サポート情報 | -- |
| }EXCOMMPORT | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| exec\_infoname[28] | | |
|  | EXECの名称情報文字列 | |
|  | | |
| exec\_version[8] | | |
|  | EXECのバージョン文字列(モジュールバージョン形式) | |
|  | | |
| df\_chipname[28] | | |
|  | デバイスファイル内の対応デバイス品種名/拡張情報の文字列 | |
|  | | |
| df\_fileversion[8] | | |
|  | デバイスファイル、ファイルバージョン文字列 | |
|  | | |
| board\_info | | |
|  | 接続エミュレータの情報 | |
|  | bit1-0 | 予約 |
|  | bit2 | エミュレータ本体種別(0:IE850/IE850A, 1:E1/E20/E2) |
|  | bit5-3 | 予約 |
|  | bit6 | スーパートレースボード有り |
|  | bit12-7 | 予約 |
|  | bit13 | EA接続有り |
|  | bit14 | TC接続有り |
|  | bit15 | ターゲット電源ON |
|  | bit31-16 | 予約 |
|  | | |
| product\_id | | |
|  | [E1/E20/E2/IE850Aの場合] | |
|  | E1/E20/E2/IE850Aのエミュレータ種別 <0x0000、0x0001、0x0002、0x0005、0x0006> | |
|  | [IE850の場合] | |
|  | TFPGA上の製品IDレジスタ値(16ビット) <0x0000 – 0xFFFF> | |
|  | | |
| product\_ver | | |
|  | [E1/E20/E2/IE850Aの場合] | |
|  | E1/E20/E2/IE850A本体に内蔵しているコントロールボードのリビジョン<0x0000 – 0xFFFF> | |
|  | [IE850の場合] | |
|  | 未定義 (0x0000固定) | |
|  | | |
| svrom\_ver | | |
|  | [E1/E20/E2/IE850Aの場合] | |
|  | コントロールボード用ファームウェア(BFW-LV0)のバージョン情報(上位4桁) | |
|  | <0x0000 – 0xFFFF> | |
|  | [IE850の場合] | |
|  | コントロールボード用ファームウェアのバージョン(ver12.34を0x1234で表す値)情報 | |
|  | <0x0000 – 0xFFFF> | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| svrom\_extraver | | | | | | |
|  | [E1/E20/E2/IE850Aの場合] | | | | | |
|  | コントロールボード用ファームウェア(BFW-LV0)のバージョン情報(下位4桁) | | | | | |
|  | <0x0000 – 0xFFFF> | | | | | |
|  | [IE850の場合] | | | | | |
|  | コントロールボード用ファームウェアのバージョン用サフィックスおよび種別情報 | | | | | |
|  | bit7-0 | コントロールボード用ファームウェアのバージョン用サフィックス | | | | |
|  |  |  | 0x00：Vバージョン(固定) | | | |
|  | bit15-8 | コントロールボード用ファームウェアの種別情報 | | | | |
|  |  |  | 0x00：V850ファミリ用CTFW | | | |
|  |  |  | 0x01：RH850ファミリ用CTFW | | | |
|  | | | | | |
| board1 | | | | | | |
|  | コートロールボード情報 | | | | | |
|  | board1.id | | | [E1/E20/E2/IE850A]の場合 | | |
|  |  | | |  | 未定義 (0x0000固定) | |
|  |  | | | [IE850]の場合 | | |
|  |  | | |  | コートロールボードID(0x0001-0x9999の番号) | |
|  | board1.ver | | | [E1/E20/E2/IE850A]の場合 | | |
|  |  | | |  | コントロールボード用ファームウェア(BFW-EML)の  バージョン情報(上位4桁) | |
|  |  | | |  | <0x0001 - 0x9999> | |
|  |  | | | [IE850]の場合 | | |
|  |  | | |  | ボードバージョン(ver12.34を0x1234で表す値)  <0x0001 - 0x9999> | |
|  | board1.fpgaver | | | [E1/E20/E2/IE850A]の場合 | | |
|  |  | | |  | FPGAバージョン(ver12.34を0x1234で表す値)  <0x0001 - 0x9999> | |
|  |  | | | [IE850]の場合 | | |
|  |  | | |  | IFPGAバージョンレジスタ下位16ビット  <0x0001 - 0x9999> | |
|  | board1.extraver | | | [E1/E20/E2/IE850A]の場合 | | |
|  |  | | |  | コントロールボード用ファームウェア(BFW-EML)の  バージョン情報(下位4桁)  <0x0001 - 0x9999> | |
|  |  | | | [IE850]の場合 | | |
|  |  | | |  | IFPGAバージョンレジスタ上位16ビット  <0x0001 - 0x9999> | |
|  | | | | | | |

|  |  |  |
| --- | --- | --- |
| board2 | | |
|  | 未使用 | |
|  | board2.id | 0x0000固定 |
|  | board2.ver | 0x0000固定 |
|  | board2.fpgaver | 0x0000固定 |
|  | board2.extraver | 0x0000固定 |
|  | | |
| board3 | | |
|  | DBボードの情報 | |
|  | [E1/E20/E2/IE850Aの場合] | |
|  | board3の情報は、すべて未使用。0で固定。 | |
|  | [IE850の場合] | |
|  | board3.id | ボードID(0x0001-0x9999の番号) |
|  | board3.ver | ボードバージョン(ver12.34を0x1234で表す値) |
|  | board3.fpgaver | TFPGAバージョンレジスタ(16ビット) |
|  | board3.extraver | TFPGAビルド番号レジスタ(16ビット) |
| board4 | | |
|  | スーパートレースボードの情報 | |
|  | [E1/E20/E2/IE850Aの場合] | |
|  | board4の情報は、すべて未使用。0で固定。 | |
|  | [IE850の場合] | |
|  | board4.id | ボードID(0x0001-0x9999の番号) |
|  | board4.ver | ボードバージョン(ver12.34を0x1234で表す値) |
|  | board4.fpgaver | SFPGAバージョン(ver12.34を0x1234で表す値) |
|  | board4.extraver | 0x0000固定 |
|  |  |  |
| board5 |  |  |
|  | 未使用 |  |
|  | board5.id | 0x0000固定 |
|  | board5.ver | 0x0000固定 |
|  | board5.fpgaver | 0x0000固定 |
|  | board5.extraver | 0x0000固定 |
|  |  |  |
| board6 |  |  |
|  | 未使用 |  |
|  | board6.id | 0x0000固定 |
|  | board6.ver | 0x0000固定 |
|  | board6.fpgaver | 0x0000固定 |
|  | board6.extraver | 0x0000固定 |
|  |  |  |
| board7 |  |  |
|  | 未使用 |  |
|  | board7.id | 0x0000固定 |
|  | board7.ver | 0x0000固定 |
|  | board7.fpgaver | 0x0000固定 |
|  | board7.extraver | 0x0000固定 |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| board8 | | | |
|  | | 未使用 | |
|  | | board8.id | 0x0000固定 |
|  | | board8.ver | 0x0000固定 |
|  | | board8.fpgaver | 0x0000固定 |
|  | | board8.extraver | 0x0000固定 |
|  | |  |  |
| board9 | | | |
|  | | 未使用 | |
|  | | board9.id | 0x0000固定 |
|  | | board9.ver | 0x0000固定 |
|  | | board9.fpgaver | 0x0000固定 |
|  | | board9.extraver | 0x0000固定 |
|  | |  |  |
| dummy[6] | | | |
|  | | 未使用(0x00固定) | |
|  | | | |
| target\_system | | | |
|  | | ターゲットシステムの種別を指定  EX\_TARGETSYS\_NORMAL ：通常のターゲットシステム  EX\_TARGETSYS\_EMUADPT ：エミュレーションアダプタを使用したターゲットシステム  上記以外 ：設定禁止 | |
|  | | | |
| trace\_lane | | | |
|  | | 外部トレースのレーン数の指定  EX\_TRACELANE\_NONE：外部トレースを使用しない  EX\_TRACELANE\_1 ：1レーンを使用する  EX\_TRACELANE\_2 ：2レーンを使用する(設定禁止、予約)  EX\_TRACELANE\_4 ：4レーンを使用する  EX\_TRACELANE\_8 ：8レーンを使用する(設定禁止、予約)  上記以外 ：設定禁止 | |
|  | | | |
| \*devfile\_inf | | | |
|  | デバイスファイル名(フルパス指定)のポインタ | | |
|  | | | |
| port1～dummy2[8] | | | |
|  | 未使用(0x00固定) | | |
|  | | | |

[Return Values]

EX\_NOERROR 正常終了

EX\_PE\_DATA 設定値が異常

EX\_FE\_DEVFILE デバイスファイルがない、または、読み込みに失敗した

EX\_FE\_COMM デバイスドライバのオープンに失敗した

EX\_SE\_M\_EMUACCFAILED エミュレータへのアクセスに失敗

EX\_FE\_ANOTHEREXECRUN ディレクトリが異なる別のEXECが既に起動している

EX\_FE\_CTFWTYPE\_ERROR エミュレータ本体に搭載されているファームウェア(CTFW)の

ファームウェア種別が正しくない。

EX\_FE\_CTFWVER\_ERROR エミュレータ本体に搭載されているファームウェア(CTFW)のバージョンが

一致しない。

EX\_FE\_IFPGATYPE\_ERROR エミュレータ本体に搭載されているFPGA(IFPGA)の種別が正しくない。

EX\_FE\_IFPGAVER\_ERROR エミュレータ本体に搭載されているFPGA(IFPGA)のバージョンが一致しない。

EX\_FE\_SFPGAVER\_ERROR 長時間トレースオプションに搭載されているFPGA(SFPGA)のバージョンが

一致しない。

EX\_FE\_TFPGATYPE\_ERROR ターゲットMCUとPODの組み合わせが正しくない。

EX\_FE\_TFPGAVER\_ERROR PODに搭載されているFPGA(TFPGA)のバージョンが一致しない。

EX\_FE\_ENVFILE ENVファイルが見つからない。

EX\_FE\_ENVDATA ENVファイルに誤りがある。

EX\_FE\_ICENUMOVER PCに接続されたICE数が規定値をオーバー

EX\_FE\_NOTFOUNDEMUADPT 対応するエミュレーションアダプタが見つからない

EX\_FE\_FANERROR IE850Aが冷却ファン停止を検出しました。

ターゲット及びIE850Aの電源をOFFしてください

EX\_FE\_VOUTERROR IE850AがVOUTからの電源供給で過電流を検出しました

EX\_FE\_BEFOREVOUTERROR IE850AがVOUTから電源供給を開始できませんでした

[Description]

[引数説明]

(1)exec\_infoname[]

ex\_getenv()のexec\_info[]の文字列からバージョンおよび日付を省略したExecutorの名称を示す文字列を返す。

文字列：“RH850G3 Executor”

(2)exec\_version[]

~~ex\_getenv()のexec\_info[]からExecutorの名称および日付の文字列を省略したバージョン情報(区切りの'.'も省略)を示す文字列を返す。~~

~~返す文字列：“Exxxxxz”または“Vxxxxx”~~

~~x　 ：実際のバージョン~~

~~z　 ：評価用Eバージョンのサフィックス~~

｢CubeSuite+ 全体システム仕様書｣に従い、モジュールバージョンの定義に変更する。

・EXECのバージョン情報を示す文字列(区切りの'.'を省略)を返す。

exec\_version[]には、V.1.23.45.67の場合、”1234567“を格納する。

※exec\_version[7]は、NULLを格納する。

(3)df\_chipname[]

デバイスファイル内の対応品種名および品種指定(-Cオプション)で指定する文字列を返す。

返す文字列：“対応品種名(品種指定)”

(4)df\_fileversion[]

デバイスファイルのバージョン情報を示す文字列(区切りの'.'を省略)を返す。

返す文字列：“Exxxxxz”または“Vxxxxx”

x：実際のバージョン

z：評価用Eバージョンのサフィックス

(5)board\_info

bit2：エミュレータ本体種別

エミュレータ本体種別の情報をビット単位で返す。

0：IE850

1：E1/E20/E2/IE850A

bit6：スーパートレースボード有り

スーパートレースボードの装着情報をビット単位で返す。

0：スーパートレースボードが装着されていない

1：スーパートレースボードが装着されている

bit13：EA接続有り

EAの接続情報をビット単位で返す。

0：EAが接続されていない

1：EAが接続されている

bit14：TC接続有り

TCの接続情報をビット単位で返す。

0：TCが接続されていない

1：TCが接続されている

bit15：ターゲット電源状態の情報を返す。

ターゲット電源状態の情報をビット単位で返す。

0：ターゲット電源がOFF

1：ターゲット電源がON

上記以外のbit：未使用であり、すべて0を返す。

(6)product\_id

【E1/E20/E2/IE850Aエミュレータの場合】

E1/E20/E2/IE850Aのエミュレータ種別を返す。

EX\_EMU\_E1 ：OCD E1 エミュレータ

EX\_EMU\_E20 ：OCD E20 エミュレータ

EX\_EMU\_E2 ：OCD E2 エミュレータ

EX\_EMU\_IE850A：FULL IE850A エミュレータ

その他：未サポート

【IE850エミュレータの場合】

TFPGA上の製品IDレジスタ値(16ビット)を返す。

(7)product\_ver

【E1/E20/E2/IE850Aエミュレータの場合】

コントロールボードのリビジョン情報を返す。

リビジョンには、Rev.A～Rev.Zを数字に置き換えて返す。(A:0x0001, B:0x0002, ,, Z:0x001A)

【IE850エミュレータの場合】

未定義 (0x0000固定)

(8)svrom\_ver

(9)svrom\_extraver

【E1/E20/E2/IE850Aエミュレータの場合】

svrom\_verとsvrom\_extraverの両方を使用して、コントロールボード上で動作するファームウェア(BFW-LV0)のバージョン情報を返す。

svrom\_ver ：ファームウェア(BFW-LV0)の上位側バージョンを返す。

svrom\_extraver：ファームウェア(BFW-LV0)の下位側バージョンを返す。

(例)[svrom\_ver=0x1234]、[svrom\_extraver=0x5678]の場合

ファームウェア(BFW-LV0)バージョン：V.1.23.45.678の意味

【IE850エミュレータの場合】

・svrom\_ver

コントロールボード用ファームウェアのバージョン(ver12.34を0x1234で表す値)情報

・svrom\_extraver

コントロールボード用ファームウェアのバージョン用サフィックスおよび種別情報

bit7-0 コントロールボード用ファームウェアのバージョン用サフィックス

0x00：Vバージョン(固定)

bit15-8 コントロールボード用ファームウェアの種別情報

0x00：V850ファミリ用CTFW

0x01：RH850ファミリ用CTFW

(10)board1

コントロールボード情報を返す。

【E1/E20/E2/IE850Aエミュレータの場合】

board1.id ：未使用。0000h固定を返す。

board1.ver ：

board1.extraver ：board1.verとboard1.extraverの両方を使用して、コントロールボード上で動作するファームウェア(BFW-LVEML)のバージョン情報を返す。

board1.ver ：ファームウェア(BFW-LVEML)の上位側バージョンを返す。

board1.extraver：ファームウェア(BFW-LVEML)の下位側バージョンを返す。

(例)[ board1.ver=0x1234]、[board1.ver=0x5678]の場合

ファームウェア(BFW-LVEML)バージョン：V.1.23.45.678の意味

board1.fpgaver ：コントロールボード上のFPGAの全体バージョン(F\_VER)を返す。

0x1234の場合、Ver12.34の意味となる。

【IE850エミュレータの場合】

board1.id ：コートロールボードID(0x0001-0x9999の番号)

board1.ver ：ボードバージョン(ver12.34を0x1234で表す値)

board1.fpgaver ：IFPGAバージョンレジスタ下位16ビット

board1.extraver ：IFPGAバージョンレジスタ上位16ビット

(11)board3

DBボードの情報

【E1/E20/E2/IE850Aエミュレータの場合】

board3の情報は、すべて未使用。0で固定。

【IE850エミュレータの場合】

board3.id ：ボードID(0x0001-0x9999の番号)

board3.ver ：ボードバージョン(ver12.34を0x1234で表す値)

board3.fpgaver ：TFPGAバージョンレジスタ(16ビット)

board3.extraver ：TFPGAビルド番号レジスタ(16ビット)

(12)board4

スーパートレースボードの情報

【E1/E20/E2/IE850Aエミュレータの場合】

board4の情報は、すべて未使用。0で固定。

【IE850エミュレータの場合】

board4.id ：ボードID(0x0001-0x9999の番号)

board4.ver ：ボードバージョン(ver12.34を0x1234で表す値)

board4.fpgaver ：SFPGAバージョン(ver12.34を0x1234で表す値)

board4.extraver ：0x0000固定

(13)\*devfile\_inf

デバイスファイル名をフルパスで指定する。

省略できない。デバイスファイルがない場合は、エラー(EX\_FE\_DEVFILE)を返す。

(14)target\_system

ターゲットシステムの種別を指定する。

設定禁止の値(EX\_TARGETSYS\_[NORMAL/EMUADPT]以外)を指定した場合は、エラー(EX\_PE\_DATA)を返す。

【E1/E20/E2/IE850エミュレータの場合】

未使用。EX\_TARGETSYS\_NORMALを指定すること。ただし、EX\_TARGETSYS\_NORMAL以外を指定してもエラーは返さない。

【IE850Aエミュレータの場合】

ターゲットにエミュレーションアダプタが接続されている場合は、EX\_TARGETSYS\_EMUADPTを指定すること。その他(FCCチップ、製品チップ)が接続されている場合は、EX\_TARGETSYS\_NORMALを指定すること。

EX\_TARGETSYS\_EMUADPTを指定したときに、\*devfile\_infで指定したデバイスファイルから対応するエミュレーションアダプタの情報を取得するが、情報が存在しなかった場合は、エラー(EX\_FE\_NOTFOUNDEMUADPT)を返す。

(15)trace\_lane

エミュレータとターゲットデバイスを接続する外部トレースI/Fのレーン数を指定する。

設定禁止の値(EX\_TRACELANE\_[NONE/1/4]以外)を指定した場合は、エラー(EX\_PE\_DATA)を返す。

【E1/E20/E2/IE850エミュレータの場合】

未使用。EX\_TRACELANE\_NONEを指定すること。ただし、EX\_TRACELANE\_NONE以外を指定してもエラーは返さない。

【IE850Aエミュレータの場合】

EX\_TRACELANE\_1、EX\_TRACELANE\_4もしくはEX\_TRACELANE\_NONEを指定すること。

ただし、実際に接続されているレーン数と異なる指定をした場合は、エラーは返さないがトレース出力先に外部トレースを選択できなくなる。

EX\_TRACELANE\_NONEを指定した場合も、同様にトレース出力先に外部トレースを選択できなくなる。

なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2 (EX\_UNIT\_TRCSEL)で行う。

※ 各デバイス種別に対応する設定値は下記の通り。

製品チップ：EX\_TRACELANE\_NONE

FCCチップ ：EX\_TRACELANE\_NONE, EX\_TRACELANE\_1

EVAチップ ：EX\_TRACELANE\_NONE, EX\_TRACELANE\_1, EX\_TRACELANE\_4

(16)board2, board[9:5], dummy[], port[1:9], dummy2[]

未使用であり、すべて0を返す。

[動作説明]

エミュレータのコンフィグレーションに必要な情報を指定し、EXEC、エミュレータ本体および各種ボードに関する情報を返す。(拡張版)

エミュレータ製品の組み合わせ(本体、ポッド、FPGA、ファームウェア)が正しくない場合は、下記[エミュレータ製品の組み合わせエラー]を通知する。

[エミュレータ製品の組み合わせエラー]

EX\_FE\_CTFWTYPE\_ERROR, EX\_FE\_CTFWVER\_ERROR, EX\_FE\_IFPGATYPE\_ERROR, EX\_FE\_IFPGAVER\_ERROR,

EX\_FE\_SFPGAVER\_ERROR, EX\_FE\_TFPGATYPE\_ERROR, EX\_FE\_TFPGAVER\_ERROR

[エミュレータ製品の組み合わせエラー]は、エミュレータハードウェアの情報とEXECで管理するバージョン情報をEXECが比較し、不一致の場合にエラーと判断する。

エミュレータ製品の組み合わせを管理するためのENVファイル(RH850Emulator.env)がEXECから

ファイルオープンできない場合は、エラー(EX\_FE\_ENVFILE)を通知する。

また、ENVファイル内の書式が正しくない場合は、エラー(EX\_FE\_ENVDATA)を通知する。

IE850Aの場合に、IE850Aの冷却ファン停止が検出されていると、エラー(EX\_FE\_FANERROR)を返す。冷却ファン停止が1回でも検出された場合、その後に冷却ファンが再稼動してもエラー(EX\_FE\_FANERROR)を返し続ける。

IE850Aの場合に、IE850Aからの供給電圧が異常値となっていた場合、エラー(EX\_FE\_VOUTERROR)を返す。

IE850Aの電源をONにした時、VOUT0またはVOUT1に0.5V以上の電圧が検出された場合も、供給電圧が異常である判断してエラー(EX\_FE\_BEFOREVOUTERROR)を返す。

IE850Aの供給電圧が異常であることを検出し、VCCLEDが消灯している状態では、IE850Aは電源供給を停止し、電源供給機能設定の変更も受け付けない。

[デバッガ関係者向け注意事項]

上記[エミュレータ製品の組み合わせエラー]に該当するエラーが発生した場合は、EXEC起動シーケンスを中断し、エラーを通知すること。

IE850Aの電源供給エラー(EX\_FE\_VOUTERROR, EX\_FE\_BEFOREVOUTERROR)および冷却ファンエラー(EX\_FE\_FANERROR)に関して、デバッグ開始時にエラーが発生した場合はデバッグ開始できないようにすること。デバッグ中にエラーが発生した場合はエラーを返すのみ行うこと。デバッグ中にエラーが発生した場合にデバッガが自動でデバッグを終了し切断しないこと。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_setenv | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **×** | **TRC1** | **／** | **TRC2** | **／** |

[Function]

デバッグ機能を含むエミュレータの環境を設定する。

[Format]

int ex\_setenv( USHORT func, EXSENV \*se )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| EXSENV | \*se | <IO> | 環境条件 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | trcmem\_size | <IO> | トレースメモリの有効トレースメッセージアドレス数 | ○ |
|  | USHORT | btrcmem\_size | <--> | -------- | -- |
|  | USHORT | brkmem\_sw | <--> | -------- | -- |
|  | UCHAR | pev\_mode | <IO> | 周辺チップ動作指定(指定可否はデバイス依存) | ○ |
|  | UCHAR | brk\_mode | <--> | -------- | -- |
|  | USHORT | flsafe\_sw | <--> | -------- | -- |
|  | UCHAR | trcp\_bksw | <--> | -------- | -- |
| }EXSENV | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |

|  |  |  |
| --- | --- | --- |
| trcmem\_size | | |
|  | EX\_M\_TRCMEM512M | 512Mトレースメッセージアドレス(IE850A時のみ) |
|  | EX\_M\_TRCMEM256M | 256Mトレースメッセージアドレス(IE850A時のみ) |
|  | EX\_M\_TRCMEM128M | 128Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM64M | 64Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM32M | 32Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM16M | 16Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM8M | 8Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM4M | 4Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ)) |
|  | EX\_M\_TRCMEM2M | 2Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM1M | 1Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ) |
|  | EX\_M\_TRCMEM512K | 512Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM256K | 256Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM128K | 128Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM64K | 64Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM32K | 32Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM16K | 16Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM8K | 8Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM4K | 4Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM2K | 2Kトレースメッセージアドレス |
|  | EX\_M\_TRCMEM1K | 1Kトレースメッセージアドレス |
| 注：1トレースメッセージアドレスにつき2フレームのトレースメッセージを格納する。 | | |
|  |  |  |
| pev\_mode | | |
|  | EX\_PEBRK | ブレーク時、周辺マクロを停止する。 |
|  | EX\_PENOBRK | ブレーク時、周辺マクロを停止させない。(default) |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_FE\_SPNOTCONNECT\_ERROR 長時間トレースオプションが装着されていない

EX\_SE\_ANOTHERPERUN 別のPEが実行中のため使用できない

EX\_DE\_TRCMEMSIZENOSPT トレースメモリの容量をサポートしていない

[Description]

[引数説明]

(1)func

操作モードを指定する。

EX\_SET(0)：seで指定する各種環境を設定する。

EX\_REF(1)：各種環境の設定値を参照する。

(2)se

各種環境条件を指定する構造体。

trcmem\_size：ICEが持つトレースメモリのうち実際に使用するフレーム数を指定する。

pev\_mode　 ：本設定の機能がある周辺マクロのみに有効であり、CPUによっては本設定の機能が有効なマクロと無効なマクロが混在する場合がある。

trcmem\_size、pev\_mode以外：未サポート。EX\_SET指定時は設定値を無視する。

[動作説明]

デバッグ機能を含むエミュレータの環境を設定する。

使用するフレーム数が大きいとトレースリード等の処理に時間がかかり、ステップ実行のレスポンスが悪くなる恐れがあり、この場合にtrcmem\_sizeによって使用するフレーム数を抑制する。デフォルトは、IE850もしくはIE850A本体が持つトレースメモリの最大とする。なお、1トレースメッセージアドレスにつき2フレームのトレースメッセージを格納する。

pev\_modeは機能をサポートしている周辺マクロに対して有効となる。デバイスによってはサポートしている周辺マクロと未サポートの周辺マクロが混在している場合がある。

IE850接続時かつSPボードがない場合にSPボード用のトレースサイズを設定した場合は、エラー(EX\_FE\_SPNOTCONNECT\_ERROR)を返す。また、IE850A接続でない場合にIE850A用のトレースサイズを設定した場合は、エラー(EX\_DE\_TRCMEMSIZENOSPT)を返す。

ユーザプログラム実行中に本関数を呼び出した場合はエラー(EX\_SE\_USRPGMRUN)を返す。ただし、ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーにはしないが設定は無視する。

[マルチコア時の注意点]

(1)物理CPU(PE)の区別なく共通となる。

[デバッガ開発者向け注意事項]

(1)ホットプラグインRUN中(接続～強制ブレークするまで)は、エラー(EX\_SE\_USRPGMRUN)を返さないが指定された設定は無視する。

(2)trcmem\_sizeに前回と異なる設定をした場合、トレース・データのクリアを行う。

[非同期デバッグモード時の注意点]

全コアがブレークしている状態のみ呼び出し可。カレントコア以外のコアがブレークしていない場合はエラー(EX\_SE\_ANOTHERPERUN)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_initimem **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

内部ROMのサイズおよび内部RAMのサイズとロケーションを設定する

[Format]

int ex\_initimem( EXIMEM \*imem )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXIMEM | \*imem | <I > | 内部ROM/RAMサイズ | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | rom\_size | <I > | 内部ROMサイズ | -- |
|  | ULONG | ram\_saadr | <I > | 内部RAM先頭アドレス | -- |
|  | ULONG | ram\_size | <I > | 内部RAMサイズ | -- |
|  | ULONG | xram\_saddr | <--> | -------- | -- |
|  | ULONG | xram\_size | <--> | -------- | -- |
| }EXIMEM | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| rom\_size | | |
|  | ------------------------ | ------------------------ |
|  |  |  |
| ram\_saddr | | |
|  | ------------------------ | ------------------------ |
|  |  |  |
| ram\_size | | |
|  | ------------------------ | ------------------------ |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_clock **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

クロックを設定/参照する。

[Format]

int ex\_clock( USHORT func, short \*clk\_type )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| USHORT | \*clk\_type | <IO> | クロックの状態 | -- |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| \*clk\_type | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_mapping **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

外部メモリのマッピングを行う。

[Format]

int ex\_mapping( EXMAP \*map )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMAP | \*map | <I > | マッピング条件 | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <I > | マッピングモード | -- |
|  | USHORT | type | <I > | マッピング属性 | -- |
|  | ULONG | s\_addr | <I > | 開始アドレス | -- |
|  | ULONG | e\_addr | <I > | 終了アドレス | -- |
| }EXMAP | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | ----- | ----- |
|  |  |  |
| type | | |
|  | ----- | ----- |
|  |  |  |
| s\_addr | | |
|  | ----- | ----- |
|  |  |  |
| e\_addr | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_pincont | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

端子のマスク機能を設定/参照する

[Format]

int ex\_pincont( USHORT func, USHORT num, ULONG \*mask\_fg )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| ULONG | \*mask\_fg | <IO> | 端子マスク指定 | ○ |
| ULONG | \*sel\_fg | <IO> | 機能選択指定 | -- |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| func | | | | |
|  | | EX\_SET | | 設定 |
|  | | EX\_REF | | 参照 |
|  | |  | |  |
| \*mask\_fg | | | | |
|  | mask\_fg[0] | | | |
|  | | bit0 | EX\_MASK\_WAIT | ウェイト信号(WAITZ)をマスクする |
|  | | bit1 | EX\_MASK\_RESET | リセット信号(RESETZ)をマスクする。(外部リセット、内部リセット共にマスクする) |
|  | | bit2 | ----- | 0固定 |
|  | | bit3 | EX\_MASK\_HWSTOP | ハードウェアストップリクエスト信号(STOPZ)をマスクする |
|  | | bit4 | EX\_MASK\_HOLD | ホールドリクエスト信号(HLDRQZ)をマスクする |
|  | | bit5-13 | ----- | 0固定 |
|  | | bit14 | EX\_MASK\_TRESET | 外部リセット(ターゲットからのリセット)をマスクする。OCDではなし。 |
|  | | bit15 | EX\_MASK\_PWRGD | PWRGD端子をマスクする。ただし、ターゲット未接続時は本設定に関係なく"マスクする"固定となる。OCDではなし。 |
|  | | bit16-31 | ----- | 0固定 |
|  | mask\_fg[1]～[8] | | | |
|  | | bit0-31 | ----- | 0固定 |
|  | |  |  |  |
| \*sel\_fg | | | | |
|  | sel\_fg[0] | | | |
|  | | bit0-31 | ----- | 0固定 |
|  | |  | |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_STANDBY\_NOSPRT　　 該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

[引数説明]

mask\_fg[0]のみ利用する。

\*sel\_fgとmask\_fg[1]～[8]の設定情報は、関数の互換のために残しているので読み出した設定を変えてはいけない。

[動作説明]

mask\_fg[0]に従い、端子のマスクや機能の設定を行う。1にすることでマスク状態になる。

EXECのデフォルトは、EXEC起動を安定に行わせるため、全端子マスク状態とする。ただし、OCDのホットプラグイン起動時は全端子マスク解除状態とする。

したがって、mask\_fg[0]のデフォルト値は下表となる。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 外部リセットマスク機能 | | あり | | なし | | 備考 |
| PWRGD端子マスク機能 | | あり | なし | あり | なし |
| OCD | 通常起動時 | － | － | － | 0x001B | － |
| ホットプラグ起動時 | － | － | － | 0x0000 | － |
| FULLエミュレータ | | 0xC01B | 0x401B | 0x801B | 0x001B | － |

ユーザプログラム実行中でも、mask\_fg[0]の端子マスクは設定できる。

ブレーク中は、EX\_MASK\_RESETの設定に関係なくリセットマスクされる。

なお、EX\_MASK\_WAIT、EX\_MASK\_HWSTOP、EX\_MASK\_HOLDに関しては、マスク対象機能であるWAIT、HWSTOP、HOLD搭載されているかどうかは製品による。対象機能が搭載していても、マスクを実際にサポートしているかどうか、さらに外部端子のみなのか内部信号も含めるのかどうかも製品によるため注意が必要。

参照(func=EX\_REF)を設定して呼び出したとき、mask\_fg[0]の設定不可bitの値は0固定となる。例えば、OCD接続時にOCDにはないEX\_MASK\_TRESET(mask\_fg[0]-bit14)を"1"に設定しても、参照時にEXECは"0"を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[マルチコア時の注意点]

PE選択状態に関係なく、すべてのPEからマスク要求を行う。

PE別々に効くのかどうかも製品によるとは思うが、V850E2系コアの時はPE別に操作しても両方に効くような構成であったこともあり、ここはEXECですべてのPEからマスク要求を行うことで挙動をそろえる意図もある。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

ユーザプログラムが実行中、初期停止を除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[デバッガ開発者向け注意事項]

(1)EXEC起動後、ユーザの指定がない場合は、ターゲット接続時では全マスク解除にするのが望ましい。マスクのままではユーザシステムに問題があっても、正常に動作してしまい誤解を招く恐れがある。ただし、FULLエミュレータのスタンドアロンの時はEX\_MASK\_TRESETをマスクしておく必要がある。

(2)一部のマスク機能は、デバイスにより実装されていないものがある。ex\_rh\_getenv – funcinfo[5]により実装有無の情報を取得できる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_m\_montimeout | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

モニタタイムアウトの時間を設定する

[Format]

int ex\_m\_montimeout( USHORT func, ULONG \*timeval )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| ULONG | \*timeval | <IO> | タイムアウト時間。単位:[秒] | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  |  |  |
| \*timeval | | |
|  | <00000000h - 0000ffffh> | タイムアウト時間を秒単位で設定する。 |
|  | 注:詳細はDescription参照 | |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

[Description]

EXECのモニタタイムアウト時間の設定参照を行う。

デフォルトは30秒。

また、0を設定した場合、タイムアウトしないでモニタの応答を待ち続ける。

[デバッガ関係者向け注意事項]

本関数で設定するタイムアウト時間は、EXEC内部処理の時間であって各EXEC関数一回の呼び出しのタイムアウト時間ではない。このため、EXEC関数によってタイムアウト時間が積算される場合がある。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_socunitinfo **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

SOCデバイス内蔵ユニットの情報を設定/参照する。

[Format]

int ex\_socunitinfo( USHORT func, USHORT unit\_type, ULONG \*unit\_info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| USHORT | unit\_type | <I > | ユニット種別 | -- |
| USHORT | \*unit\_info | <IO> | ユニット別の情報 | -- |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| unit\_type | | |
|  | ----- | ----- |
|  |  |  |
| \*unit\_unfo | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。ex\_socunitinfo2を使用すること。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_socunitinfo2 | **SYNC** | **EMU** | **注1** | **BRK** | **注1** | **TRC** | **注1** | **TIM** | **注1** | **COV** | **注1** | **HOT** | **注1** |
| **ASYNC** | **EMU1** | **注1** | **EMU2** | **注1** | **BRK1** | **注1** | **BRK2** | **注1** | **TRC1** | **注1** | **TRC2** | **注1** |

[Function]

SOCデバイス内蔵ユニットの情報を設定/参照する。拡張版。

注1：指定するunit\_typeごとに、EXECの状態と関数呼び出し可否が違ってくる。EXEC状態と関数呼び出し可否の関係については、ex\_socunitinfo2の章の最終ページに記載されている表を参照すること。

[Format]

int ex\_socunitinfo2( USHORT func, USHORT unit\_type, USHORT \*num, ULONG \*unit\_info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | unit\_type | <I > | ユニット種別 | ○ |
| USHORT | \*num | <IO> | ユニット別の情報の個数 | ○ |
| ULONG | \*unit\_info | <IO> | ユニット別の情報 | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  |  |  |
| unit\_type | | |
|  | EX\_UNIT\_LPDOPT | LPD接続の設定 |
|  | EX\_UNIT\_RSU | OCDセキュリティIDコードの解除/参照 |
|  | EX\_UNIT\_CG | クロック関係の設定 |
|  | EX\_UNIT\_FLNUMINFO | 未サポート(コードフラッシュブロック個数情報取り出し) |
|  | EX\_UNIT\_FLFUNCINFO | コードフラッシュ機能情報参照 |
|  | EX\_UNIT\_FLMAPINFO | コードフラッシュマッピング情報参照 |
|  | EX\_UNIT\_FLBLOCKINFO | コードフラッシュブロックアドレス情報取り出し |
|  | EX\_UNIT\_DFLNUMINFO | データフラッシュブロック個数情報取り出し |
|  | EX\_UNIT\_DFLMAPINFO | データフラッシュマッピング情報参照 |
|  | EX\_UNIT\_DFLBLOCKINFO | データフラッシュブロックアドレス情報取り出し |
|  | EX\_UNIT\_DFLAREABLOCKINFO | 領域別データフラッシュブロックアドレス情報取り出し |
|  | EX\_UNIT\_DCUSEC | OCDセキュリティIDコードの設定  注：本開発バージョンではサポートしない |
|  | EX\_UNIT\_FLMKOP | フラッシュマスクオプション設定  注：RH850 G4コア世代MCUではサポートしない。代わりに  　　フラッシュエキストラエリアの書き換えはex\_memwriteで  サポートする。 |
|  | EX\_UNIT\_FLSECINFO | コードフラッシュのExtra領域(SecurityFlag)の設定/参照  注：本開発バージョンではサポートしない |
|  | EX\_UNIT\_FLEXTRAINIT | コードフラッシュのExtra(フラッシュマスクオプション、SecurityFlag、LockBit)領域の初期化  注：本開発バージョンではサポートしない |
|  | EX\_UNIT\_FLLBINFO | コードフラッシュのExtra領域(LockBit)の設定/参照  注：RH850 G4コア世代MCUではLockBitがないためサポート  しない。BlockProtectionAreaの読み書きはex\_memreadと  ex\_memwriteでサポートする。 |
|  | EX\_UNIT\_DFLBLNKINFO | データフラッシュブランク状態参照 |
|  | EX\_UNIT\_FID | フラッシュメモリ用セキュリティIDコードの認証/  解除状態の参照(G3世代MCUのみ対応) |

|  |  |  |
| --- | --- | --- |
|  | EX\_UNIT\_CMFID | 起動時に必ず認証が必要IDコード(Customer ID/Data Flash ID)の認証/認証解除状態の参照(G4世代のみ対応) |
|  | EX\_UNIT\_OPTID | 任意で認証するIDコード(Customer ID/Data Flash ID以外)の認証/認証解除状態の参照(G4世代のみ対応) |
|  | EX\_UNIT\_COREPESEL | 操作対象PE(NM)/VM/HT番号の設定/参照 |
|  | EX\_UNIT\_SWBRKFUNCTION | フラッシュメモリへのソフトウェアブレーク設定方式の設定 |
|  | EX\_UNIT\_EXTFLINFO | 未使用(外付けフラッシュ情報) |
|  | EX\_UNIT\_RUNCONTOPT | ex\_run()の状態継続の設定 |
|  | EX\_UNIT\_DMMRRMOPT | DMM、RRMの動作設定 |
|  | EX\_UNIT\_FLPGOPT | フラッシュメモリへの書き込み保護動作関連の設定 |
|  | EX\_UNIT\_TRCSEL | 使用するトレース出力タイプの設定/参照 |
|  | EX\_UNIT\_SLAVEIDSEL | 操作対象スレーブIDの設定/参照 |
|  | EX\_UNIT\_VIRTUALINFO | 仮想化支援機能使用有無の設定/参照  (G3世代のみ対応) |
|  | EX\_UNIT\_FLCCROPT | フラッシュメモリ書き換え時のクロック制御設定 |
|  | EX\_UNIT\_ASYNCPEOPT | 複数PE間の非同期デバッグモードの設定/参照 |
|  | EX\_UNIT\_TRACEOPT | トレース関連のオプション設定 |
|  | EX\_UNIT\_TRACEMSGPEID | トレースメッセージに記録されるPEIDの値を設定/参照 |
|  | EX\_UNIT\_WORKRAM | ワークRAM領域の開始アドレスを設定/参照  ワークRAM領域のサイズを参照 |
|  | EX\_UNIT\_DBGINFOUT | 任意のデバッグ情報の設定/参照 |
|  | EX\_UNIT\_CLUSTERSEL | 操作対象クラスタ番号の設定/参照(G4世代のみ対応) |
|  | EX\_UNIT\_EXTTRACERATE | 外部トレースの転送速度の設定/参照(IE850A接続時のみ対応) |
|  | EX\_UNIT\_SVR | SVRパラメータの設定(G4世代のみ対応) |
|  | EX\_UNIT\_EFRECOVERY | SWASリカバーコマンド実行(G4世代のみ対応) |
|  | EX\_UNIT\_HEGESEL | 仮想化支援機能使用時のデバッグ対象となるホストモード、ゲストモード、ブレーク機能オプション、ステップ機能オプションの設定/参照  (G4世代のみ対応) |
|  | EX\_UNIT\_INITSTOPDBG | 初期停止・スタンバイモードデバッグ環境の設定/参照 |

|  |  |  |
| --- | --- | --- |
| \*num | | |
|  | <0000h – FFFFh> | unit\_infoの個数を設定する。 |
|  | 注：参照時は返したunit\_info数を返す。指定したユニットに足りるunit\_info数の領域があるとこが前提。なお、参照時でもすべてのユニットで、\*numは設定して呼び出すこと。 | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| \*unit\_info | | | |
|  | <EX\_UNIT\_LPDOPT> (num=12) | | |
|  | unit\_info[0] | LPD接続の通信モードの指定(OCDのみ) | |
|  |  | 0 | 1ピンモード |
|  |  | 3 | 4ピンモード(デフォルト) |
|  | unit\_info[1] | ビット単位でオプションバイトの処理に関する指定(OCDのみ) | |
|  |  | bit0 | 起動時ピンモードが異なっているときの処理を指定 |
|  |  |  | 0：エラーにして接続中止する(デフォルト) |
|  |  |  | 1：オプションバイトを指定ピンモードに書き変えて接続する |
|  |  | bit1 | 終了時、オプションバイトのピンモードの処理の指定 |
|  |  |  | 0：そのままの状態で終了(デフォルト) |
|  |  |  | 1：オプションバイトを初期状態(JTAG)に書き換えて終了する |
|  |  | bit2-3 0固定 | |
|  |  | bit4 | 起動時、STMSELの変更に関する指定 |
|  |  |  | 0：変更しない(デフォルト) |
|  |  |  | 1：0(ユーザブートモード)にする |
|  |  | bit5-27 0固定 | |
|  |  | bit28 | 起動時、オプションバイトの処理の抑制 |
|  |  |  | 0：オプションバイト処理を行う(デフォルト) |
|  |  |  | 1：オプションバイト処理を行わない |
|  |  | bit29-31 0固定 | |
|  | unit\_info[2] | LPDクロック周波数(4pin)またはボーレート(1pin)の指定 | |
|  |  | 1以上の場合、エミュレータで設定可能な設定値に近い値で接続する  値の目安は以下の通りだが、任意な値を設定可能 | |
|  |  | OCD 4pin ： 16500、11000、5500、3000 KHz(F1Aは12MHz以下)※ | |
|  |  | OCD 1pin ： 2000、1000、500 Kbps | |
|  |  | IE850 ： 20000、10000 KHz | |
|  |  | ※E2/IE850Aエミュレータでは上記に加えて以下の周波数も設定可能  　40000、33000、25000、20000 KHz | |
|  |  | 0の場合、下記に示す接続処理をする(デフォルト)  OCD 4pin ： 16500KHz(PFC1A、PFC1B)、11000KHz(F1A)※  OCD 1pin ： MCUのデフォルト設定のボーレート(かなり遅い)  IE850 ： 20000KHz  ※他の品種についてはF/W機能仕様書を参照すること | |
|  | unit\_info[3] | LPDクロック周波数(4pin)またはボーレート(1pin)の決定値  (接続後に参照のみ可能) | |
|  |  | 単位は、KHz(4pin)またはKbps(1pin) | |
|  | unit\_info[4] | デバッグクロックの決定値(接続後に参照のみ可能) | |
|  |  | 単位は、KHz | |
|  | unit\_info[5] | 実際に行ったオプションバイト処理の参照(参照のみ) | |
|  |  | bit0 | 起動でピンモードを書き換えたかどうかの情報 |
|  |  |  | 0：書き換えていない |
|  |  |  | 1：書き換えた |
|  |  | bit1-7 | 0固定 |
|  |  | bit8-15 | 読み出したときのピンモードの状態 |
|  |  |  | 0：1ピンモード |
|  |  |  | 3：4ピンモード |
|  |  |  | F：初期状態(JTAG) |
|  |  |  | FF：読み出していない |
|  |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | unit\_info[6] | | ビット単位で起動に関するオプションの指定 | | | | | |
|  |  | | bit0 | | | 起動時のRAM初期化に関する指定 | | |
|  |  | |  | | | 0：RAM初期化を行う(デフォルト) | | |
|  |  | |  | | | 1：RAM初期化を行わない | | |
|  |  | | bit1 | | | CANブートデバッグに関する指定 | | |
|  |  | |  | | | 0：CANブートデバッグを行わない(デフォルト) | | |
|  |  | |  | | | 1：CANブートデバッグを行う | | |
|  |  | | bit2－31 | | | 0固定 | | |
|  | unit\_info[7] | | コールドスタートシーケンスの手順指定 | | | | | |
|  |  | | 0 | | | 端子リセットディレイありPiggyBoardのウェイトを行うシーケンス(デフォルト) | | |
|  |  | | 1 | | | BIST搭載デバイスの通常シーケンス | | |
|  |  | | 注：デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合のみ有効となる機能であるため、bit11(RESET\_DELAY)が0の場合には、unit\_info[7]の指定は無視される。 | | | | | |
|  | unit\_info[8] | | デバッグするマップモードを指定 | | | | | |
|  |  | | 0 | | | シングルマップモード(デフォルト) | | |
|  |  | | 1 | | | ダブルマップモード | | |
|  |  | | 2 | | | E2x-FCC2コンパチブルモード(デフォルト) | | |
|  |  | | 注：シングルマップモードとダブルマップモード両方に対応しているデバイスのみ有効となる機能である。ダブルマップモードに対応していないデバイスの場合、ダブルマップモードを指定した場合はEX\_UNIT\_RSUにてEX\_DE\_SINGLEMAPONLYエラーが返る。  また、E2x-FCC2コンパチブルモードを対応しているデバイスではE2x-FCC2コンパチブルモードがデフォルトとなる。E2x-FCC2コンパチブルモードを対応していないデバイスで指定した場合、EX\_UNIT\_RSU発行時にエラー応答(EX\_DE\_NOSPRT\_SELMAPMODE)する。 | | | | | |
|  | unit\_info[9]～[11] | | 予備 | | | | | |
|  |  | | 0固定 | | |  | | |
|  |  | |  | | |  | | |
|  | | <EX\_UNIT\_RSU> (num=5, num=9) | | | | | |
|  | | セキュリティIDが128ビットの場合(num=5) | | | | | |
|  | | unit\_info[0] | | セキュリティIDコード0 (MCUレジスタ並び b0-b31) | | | |
|  | | unit\_info[1] | | セキュリティIDコード1 (MCUレジスタ並び b32-b63) | | | |
|  | | unit\_info[2] | | セキュリティIDコード2 (MCUレジスタ並び b64-b95) | | | |
|  | | unit\_info[3] | | セキュリティIDコード3 (MCUレジスタ並び b96-b127) | | | |
|  | | unit\_info[4] | | 解除結果(1ならば解除成功、1以外は解除失敗) | | | |
|  | | セキュリティIDが256ビットの場合(num=9) | | | | | |
|  | | unit\_info[0] | | セキュリティIDコード0 (MCUレジスタ並び b0-b31) | | | |
|  | | unit\_info[1] | | セキュリティIDコード1 (MCUレジスタ並び b32-b63) | | | |
|  | | : | | : | | | |
|  | | unit\_info[7] | | セキュリティIDコード7 (MCUレジスタ並び b224-b255) | | | |
|  | | unit\_info[8] | | 解除結果(1ならば解除成功、1以外は解除失敗) | | | |
|  | |  | |  | | | |
|  | | <EX\_UNIT\_CG> (num=4) | | | | | |
|  | | unit\_info[0] | | 00000000h固定 | | | |
|  | | unit\_info[1] | | メインクロック(EXTAL/XTAL)に入力している周波数(単位：kHz) | | | |
|  | | unit\_info[2] | | サブクロックに入力している周波数(単位：Hz)(0000000h固定) | | | |
|  | | unit\_info[3] | | ユーザが設定するPLL逓倍数 | | | |
|  | |  | | | | | |
|  | | <EX\_UNIT\_FLFUNCINFO> (num=4) | | | | | |
|  | | unit\_info[0] | | コードフラッシュの領域分割対応情報 | | | |
|  | |  | | 0：領域分割対応無 | | | |
|  | |  | | 1：領域分割対応有 | | | |
|  | | unit\_info[1] | | コードフラッシュの領域数 | | | |
|  | | unit\_info[2] | | FLMD端子の自動切り換え情報。 | | | |
|  | |  | | 0：R.F.U(FLMDなし等) | | | |
|  | |  | | 1：OCDエミュレータによる操作可能 | | | |
|  | |  | | 2：Low固定 | | | |
|  | |  | | 3：High固定 | | | |
|  | |  | | 4：周辺IOレジスタによる内蔵プルアップ操作可能 | | | |
|  | | unit\_info[3] | | フラッシュマクロ種別情報 | | |
|  | | <EX\_UNIT\_FLMAPINFO> (num=n\*2:nはEX\_UNIT\_FLFUNCINFOの領域数) | | | | | |
|  | | unit\_info[0] | | 領域1の先頭アドレス | | | |
|  | | ： | | ： | | | |
|  | | unit\_info[n-1] | | 領域nの先頭アドレス | | | |
|  | | unit\_info[n] | | 領域1のブロック数 | | | |
|  | | ： | | ： | | | |
|  | | unit\_info[2n-1] | | 領域nのブロック数 | | | |
|  | |  | | | | | |
|  | | <EX\_UNIT\_FLBLOCKINFO> (num=n:EX\_UNIT\_FLMAPINFOの領域で得たブロック数 + 1) | | | | | | |
|  | | unit\_info[0] | | | 参照する領域の番号 | | | |
|  | | unit\_info[1] | | | ブロック0のエンドアドレス | | | |
|  | | unit\_info[2] | | | ブロック1のエンドアドレス | | | |
|  | | ： | | | ： | | | |
|  | | unit\_info[n] | | | ブロックnのエンドアドレス | | | |
|  | |  | | |  | | | |
|  | | <EX\_UNIT\_DFLNUMINFO> (num=3) | | | | | | |
|  | | unit\_info[0] | | | データフラッシュの領域1のブロック数。0の場合は、データフラッシュ情報がない | | | |
|  | | unit\_info[1] | | | データフラッシュの領域分割対応情報  ※領域分割対応有の場合は、ブロック数をEX\_UNIT\_DFLMAPINFOで参照する必要がある | | | |
|  | |  | | | 0：領域分割対応無 | | | |
|  | |  | | | 1：領域分割対応有 | | | |
|  | | unit\_info[2] | | | データフラッシュの領域数 | | | |
|  | |  | | | | | | |
|  | | <EX\_UNIT\_DFLMAPINFO> (num=n\*2:nはEX\_UNIT\_DFLNUMINFOの領域数) | | | | | | |
|  | | unit\_info[0] | | | 領域1の先頭アドレス | | | |
|  | | ： | | | ： | | | |
|  | | unit\_info[n-1] | | | 領域nの先頭アドレス | | | |
|  | | unit\_info[n] | | | 領域1のブロック数 | | | |
|  | | ： | | | ： | | | |
|  | | unit\_info[2n-1] | | | 領域nのブロック数 | | | |
|  | |  | | | | | | |
|  | | <EX\_UNIT\_DFLBLOCKINFO> (num=n:EX\_UNIT\_DFLNUMINFOで得たブロック数) | | | | | | |
|  | | unit\_info[0] | | | ブロック0のエンドアドレス | | | |
|  | | unit\_info[1] | | | ブロック1のエンドアドレス | | | |
|  | | ： | | | ： | | | |
|  | | unit\_info[n-1] | | | ブロックn-1のエンドアドレス | | | |
|  | |  | | |  | | | |
|  | | <EX\_UNIT\_DFLAREABLOCKINFO> (num=n:EX\_UNIT\_DFLMAPINFOの領域で得たブロック数 + 1) | | | | | | |
|  | | unit\_info[0] | | | 参照する領域の番号 | | | |
|  | | unit\_info[1] | | | ブロック0のエンドアドレス | | | |
|  | | unit\_info[2] | | | ブロック1のエンドアドレス | | | |
|  | | ： | | | ： | | | |
|  | | unit\_info[n-1] | | | ブロックn-1のエンドアドレス | | | |
|  | |  | | |  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <EX\_UNIT\_DCUSEC> (num=5) | | |
|  | unit\_info[0] | 設定結果(1ならば設定成功、1以外は設定失敗) | |
|  | unit\_info[1] | セキュリティIDコード0 | |
|  | unit\_info[2] | セキュリティIDコード1 | |
|  | unit\_info[3] | セキュリティIDコード2 | |
|  | unit\_info[4] | セキュリティIDコード3 | |
|  |  | | |
|  | 注：本開発バージョンではサポートしない | | |
|  |  | | |
|  | <EX\_UNIT\_FLMKOP> (num=8) | | |
|  | unit\_info[0] | フラッシュマスクオプションデータ1 | |
|  | unit\_info[1] | フラッシュマスクオプションデータ2 | |
|  | unit\_info[2] | フラッシュマスクオプションデータ3 | |
|  | unit\_info[3] | フラッシュマスクオプションデータ4 | |
|  | unit\_info[4] | フラッシュマスクオプションデータ5 | |
|  | unit\_info[5] | フラッシュマスクオプションデータ6 | |
|  | unit\_info[6] | フラッシュマスクオプションデータ7 | |
|  | unit\_info[7] | フラッシュマスクオプションデータ8 | |
|  |  | | |
|  | 注：未サポートのデータには0xFFFFFFFFを返す。詳細はDescription参照 | | |
|  |  | | |
|  | <EX\_UNIT\_FLSECINFO> (num=2) | | |
|  | unit\_info[0] | 未使用(0固定) | |
|  | unit\_info[1] | ビット単位でSecurityFlagを設定。 | |
|  |  | bit0 | 0：ブロック消去許可  1：ブロック消去禁止 |
|  |  | bit1 | 0：ライト許可  1：ライト禁止 |
|  |  | bit2 | 0：リード許可  1：リード禁止 |
|  |  | bit31-3：未使用(0固定) | |
|  |  | | |
|  | 注：SecurityFlagはIE850のみ使用可能。本開発バージョンではサポートしない。 | | |
|  |  | | |
|  | <EX\_UNIT\_FLEXTRAINIT> (num=1) | | |
|  | unit\_info[0] | 77777777h | Extra領域(セキュリティIDコード、フラッシュマスクオプション、SecurityFlag)の初期化を行う |
| 77777777h以外 | 何もしない |
|  |  | | |
|  | 注：EX\_SETのみ。IE850のみ使用可能。本開発バージョンではサポートしない。 | | |
|  |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | <EX\_UNIT\_FLLBINFO> | | |
|  | [EX\_SET時] (num=3) | | | |
|  | | unit\_info[0] | | 指定ブロック範囲のLockBit状態設定 |
|  | |  | | 0：LockBit=0(プロテクト状態) |
|  | |  | | 1：LockBit=1(非プロテクト状態) |
|  | | unit\_info[1] | | 指定範囲の開始ブロック番号 |
|  | | unit\_info[2] | | 指定範囲の終了ブロック番号 |
|  |  | | | |
|  | [EX\_REF時] (num=3+n：nは参照するブロック数)  ※全ブロックのLockBit情報を参照する場合、nはEX\_UNIT\_FLMAPINFOによって得られるブロック数の合計になる。 | | | |
|  | | unit\_info[0] | | 未使用。0が返る。 |
|  | | unit\_info[1] | | 指定範囲の開始ブロック |
|  | | unit\_info[2] | | 指定範囲の終了ブロック |
|  | | unit\_info[3+0] | | 指定範囲の開始ブロックのLockBit情報 |
|  | | ： | | ： |
|  | | unit\_info[3+m] | | 指定範囲の終了ブロックのLockBit情報 |
|  | |  | | m：終了ブロック - 開始ブロック  単一ブロックの場合、unit\_info[3]のみとなる。 |
|  | |  | |  |
|  | | 注：詳細はDescription参照 | | |
|  | |  | | |
|  | | <EX\_UNIT\_DFLBLNKINFO> (num=3+n)  ※nはブランクチェック結果が返る数。 unit\_info[0]=0の場合、参照するブロック数で、最大はEX\_UNIT\_DFLMAPINFOで得たブロック数の合計となる。このため、nは1以上、ブロック数の合計未満となる。 unit\_info[0]=1の場合、参照するワード数で(RV40Fは1ワード単位書き込みなので)、最大は16となる(1ブロック分)。 | | |
|  | | unit\_info[0] | | ブランクチェックモード |
|  | |  | | 0：ブロック単位 |
|  | |  | | 1：書き込みサイズ単位 |
|  | | unit\_info[1] | | ブランクチェック開始ブロック/アドレス |
|  | | unit\_info[2] | | ブランクチェック終了ブロック/アドレス |
|  | | unit\_info[3+0] | | ブランクチェック開始ブロック/アドレスのブランク情報 |
|  | | ： | | ： |
|  | | unit\_info[3+m] | | ブランクチェック終了ブロック/アドレスのブランク情報 |
|  | |  | | 0：ブランク |
|  | |  | | 1：ブランク以外 |
|  | | 注： | unit\_info[3+0] - unit\_info[3+m]にブランク情報としてブランクチェックの結果が返る。  mはunit\_info[0]の設定値により、  unit\_info[0]=0：m＝終了ブロック - 開始ブロック  unit\_info[0]=1：m＝(終了アドレス - 開始アドレス)/書き込みサイズ  ただし、単一ブロックの場合、unit\_info[3]のみとなる。 | |
|  | | 詳細はDescription参照 | | |
|  | |  | | |

|  |  |  |
| --- | --- | --- |
|  | <EX\_UNIT\_FID>(num=18) | |
|  | unit\_info[0] | CodeFlashセキュリティIDコード0  (MCUレジスタ並び b0-b31) |
|  | unit\_info[1] | CodeFlashセキュリティIDコード1  (MCUレジスタ並び b32-b63) |
|  | : | : |
|  | unit\_info[7] | CodeFlashセキュリティIDコード7  (MCUレジスタ並び b224-b255) |
|  | unit\_info[8] | CodeFlashセキュリティID解除結果  (1ならば解除成功、1以外は解除失敗) |
|  | unit\_info[9] | DataFlashセキュリティIDコード0  (MCUレジスタ並び b0-b31) |
|  | unit\_info[10] | DataFlashセキュリティIDコード1  (MCUレジスタ並び b32-b63) |
|  | : | : |
|  | unit\_info[16] | DataFlashセキュリティIDコード7  (MCUレジスタ並び b224-b255) |
|  | unit\_info[17] | DataFlashセキュリティID解除結果  (1ならば解除成功、1以外は解除失敗) |
|  |  |  |
|  | <EX\_UNIT\_CMFID>(num=11) | |
|  | unit\_info[0] | Customer ID/Data Flash IDに対するIDコードタイプ |
|  | unit\_info[1] | IDコードタイプごとのIDコード番号 |
|  | unit\_info[2] | unit\_info[1:0]で指定したIDコードタイプ/IDコード番号に対応したIDコード0(MCUレジスタ並び b0-b31) |
|  | unit\_info[3] | unit\_info[1:0]で指定したIDコードタイプ/IDコード番号に対応したIDコード1(MCUレジスタ並び b32-b63) |
|  | : | : |
|  | unit\_info[9] | unit\_info[1:0]で指定したIDコードタイプ/IDコード番号に対応したIDコード7(MCUレジスタ並び b224-b255) |
|  | unit\_info[10] | unit\_info[1:0]で指定したIDコードタイプ/IDコード番号に対応したIDコード認証の解除結果(1ならば解除成功、1以外は解除失敗) |
|  |  |  |
|  | <EX\_UNIT\_OPTID>(num=10) | |
|  | unit\_info[0] | 任意で認証するIDコード番号 |
|  | unit\_info[1] | unit\_info[0]で指定したIDコード番号に対応したIDコード0  (MCUレジスタ並び b0-b31) |
|  | unit\_info[2] | unit\_info[0]で指定したIDコード番号に対応したIDコード1  (MCUレジスタ並び b32-b63) |
|  | : | : |
|  | unit\_info[8] | unit\_info[0]で指定したIDコード番号に対応したIDコード7  (MCUレジスタ並び b224-b255) |
|  | unit\_info[9] | unit\_info[0]で指定したIDコード番号に対応したIDコード認証の解除結果(1ならば解除成功、1以外は解除失敗) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | <EX\_UNIT\_COREPESEL> | | | | |
|  | [RH850G3] | | | | |
|  | 仮想化支援機能使用有無の設定により、numに設定する個数を合わせる必要がある。 | | | | |
|  | 仮想化支援機能使用有無の設定は、unit\_type=EX\_UNIT\_VIRTUALINFOで行う。 | | | | |
|  | [RH850G4] | | | | |
|  | 仮想化支援機能使用有無に関わらず、num=1とする。 | | | | |
|  |  |  | | | |
|  | [仮想化支援機能を使用しない場合(num=1)] | | | | |
|  | unit\_info[0] | CPU個別リソース操作対象のPE番号 | | | |
|  |  |  | | | |
|  | [仮想化支援機能を使用する場合(num=3)] | | | | |
|  | unit\_info[0] | CPU個別リソース操作対象のPE番号 | | | |
|  | unit\_info[1] | CPU個別リソース操作対象のマシンもしくはスレッド番号 | | | |
|  |  | 0-7 | ：VMを選択する場合に設定可能な番号 | | |
|  |  | 0-31 | ：HTを選択する場合に設定可能な番号 | | |
|  |  | 0xFF | ：NMを選択する場合に設定する番号 | | |
|  |  |  | | | |
|  | unit\_info[2] | CPU個別リソース操作対象のマシンもしくはスレッドタイプ | | | |
|  |  | EX\_COREPESEL\_TYPE\_NM | | | ：NMを選択 |
|  |  | EX\_COREPESEL\_TYPE\_VM | | | ：VMを選択 |
|  |  | EX\_COREPESEL\_TYPE\_HT | | | ：HTを選択 |
|  |  | | | | |
|  | 注：仮想化支援機能有無によりnumの個数が決まる。詳細は[Description]を参照。 | | | | |
|  | | | | | |
|  | <EX\_UNIT\_SWBRKFUNCTION> (num=2) | | | | |
|  | unit\_info[0] | フラッシュメモリエリアに対するソフトウェアブレークタイプの設定可否情報(参照のみ) | | | |
|  |  | bit0 | | 未使用(0固定) | |
|  |  | bit1 | | 0：BRK命令置き換えソフトウェアブレークの設定不可  1：BRK命令置き換えソフトウェアブレークの設定可能  (BRK命令置き換えソフトウェアブレークのみのサポートになるため「1」固定) | |
|  |  | bit2-31 | | 未使用(0固定) | |
|  | unit\_info[1] | 使用するフラッシュメモリエリアに対するソフトウェアブレークタイプの選択(設定/参照) | | | |
|  |  | bit0 | | 未使用(0固定) | |
|  |  | bit1 | | 0：BRK命令置き換えソフトウェアブレークを使用しない  1：BRK命令置き換えソフトウェアブレークを使用する  (BRK命令置き換えソフトウェアブレークのみのサポートになるため「1」固定) | |
|  |  | bit2-31 | | 未使用(0固定) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | <EX\_UNIT\_RUNCONTOPT> (num=1) | | | |
|  | unit\_info[0] | ビット単位でユーザプログラム実行前状態継続の指定を行う | | |
|  |  | bit0 | | 0：シーケンシャルイベントの段数とパスカウントの状態を初期化(デフォルト)  1：シーケンシャルイベントの段数とパスカウントの状態を継続 |
|  |  | bit1 | | 0：ブレーク以前にトレース終了状態になっていてもトレースを開始する(デフォルト)  1：ブレーク以前にトレース終了状態になっていた場合はトレースを開始させない |
|  |  | bit2 | | 0：ブレーク前セクション・トレース最中の時でもセクション開始待ちに仕切りなおして再開する(デフォルト)  1：ブレーク前セクション・トレース最中の時はセクション中を継続して再開する |
|  |  | bit3 | | 0：トレースディレイトリガが発生してディレイカウント最中の場合でもディレイトリガ発生待ちに初期化(デフォルト)  1：トレースディレイトリガが発生してディレイカウント最中の場合はトレース開始させない |
|  |  | bit4 | | 0：ユーザプログラム実行再開時にブレーク前のRUN-BREAKタイマの計測結果を初期化する(デフォルト)  1：ユーザプログラム実行再開時にブレーク前のRUN-BREAKタイマの計測結果を初期化しないで蓄積する |
|  |  | ~~bit5~~ | | ~~0：パフォーマンスタイマが計測停止中の場合でも、ユーザプログラム実行再開で計測も再開する(デフォルト)~~  ~~1：パフォーマンスタイマが計測停止中の場合、ユーザプログラム実行を再開しても計測は再開させない~~ |
|  | | | | |
|  | <EX\_UNIT\_DMMRRMOPT> (num=1) | | | |
|  | unit\_info[0] | | ビット単位でDMM、RRMの動作設定を行う | |
|  |  | | bit0-31 | 0固定 |
|  |  | | | |
|  | <EX\_UNIT\_FLPGOPT> (\*num=1) | | | |
|  | unit\_info[0] | | ビット単位でEXECでの書き込み保護関連の動作設定を行う。 | |
|  |  | | bit0-5 | 未使用(0固定) |
|  |  | | bit6 | 0：コードフラッシュのセルフプログラミングを行う。  1：コードフラッシュのセルフプログラミングを行わない。  (デフォルト) |
|  |  | | bit7-31 | 未使用(0固定) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | <EX\_UNIT\_TRCSEL> (num=3) | | | | | |
|  | unit\_info[0] | 予約  ~~トレース出力タイプの設定可否およびトレース機能(TCU)の実装有無の情報(参照のみ)~~ | | | | |
|  |  | bit0 | | 未使用(0固定)  ~~0：外部トレースの設定不可~~  ~~1：外部トレースの設定可能~~ | | |
|  |  | bit1 | | 未使用(0固定)  ~~0：内蔵トレースの設定不可~~  ~~1：内蔵トレースの設定可能~~ | | |
|  |  | bit2 | | 未使用(0固定)  ~~0：CPUトレース機能(TCU\_CPU)は未実装~~  ~~1：CPUトレース機能(TCU\_CPU)は実装~~ | | |
|  |  | bit3 | | 未使用(0固定)  ~~0：CPUスレーブトレース機能は未実装~~  ~~1：CPUスレーブトレース機能は実装~~ | | |
|  |  | bit[31:4] | | 未使用(0固定) | | |
|  | unit\_info[1] | トレース出力タイプの選択(設定/参照) | | | | |
|  |  | bit0 | | 0：外部トレースを使用しない  1：外部トレースを使用する | | |
|  |  | bit1 | | 0：内蔵トレースを使用しない  1：内蔵トレースを使用する | | |
|  |  | bit2 | | 0：外部ソフトトレース単線出力を使用しない  1：外部ソフトトレース単線出力を使用する | | |
|  |  | bit[31:3] | | 未使用(0固定) | | |
|  |  | 設定のDefaultは、下記の通りとする。  IE850接続時：  bit0=1, bit1=0, bit2=0 (外部トレースを選択する)  IE850接続時以外(E1/E20/E2/IE850A接続時)：※1※2  bit0=0, bit1=1, bit2=0 (内蔵トレースを選択する)  ※1 内蔵トレースが未実装の場合は、bit1=0となる。  ※2 IE850A接続時は、外部トレースが使用可能な場合でも、IE850とは異なり、デフォルトの選択は内蔵トレースとなる。 | | | | |
|  | unit\_info[2] | ~~トレースメモリサイズの情報(参照のみ)~~  ~~(単位：バイト)~~  未使用(0固定) | | | | |
|  | | | | | | |
|  | <EX\_UNIT\_SLAVEIDSEL> (num=1) | | | | | |
|  | unit\_info[0] | 0x00000000～0xFFFFFFFF  マスタ資源もしくはスレーブ資源に対応するスレーブID番号。  以下にスレーブIDの定義を示す。 | | | | |
|  |  |  | スレーブID値 | | |  |
|  |  |  | 0x00000000※1 | | CPU資源 |  |
|  |  |  | 0x00000001 | | EMS資源 |  |
|  |  |  | 0x00000002※2,※3 | | GRM/CRM資源 |  |
|  |  |  | 0x00000003※4 | | GVC/KVC/AXI/DMA資源 |  |
|  |  |  | 0x00000004※5 | | LRM資源 |  |
|  |  | ※1 操作対象のTEU\_CPUまたはTCU\_CPUは、EX\_UNIT\_COREPESELで選択しているPEとなる。  ※2 GRM, CRMは排他実装となる。  ※3 CRMが実装されている場合、操作対象のTEU\_CRMまたはTCU\_CRMは、EX\_UNIT\_CLUSTERSELで選択しているクラスタ番号となる。  ※4 GVC, KVC, AXI, DMAは排他実装となる。  ※5 LRMは各PEに実装されている。操作対象のTEU\_LRMまたはTCU\_LRMはEX\_UNIT\_COREPESELで選択しているPEとなる。 | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <EX\_UNIT\_VIRTUALINFO> (num=1) | | |
|  | unit\_info[0] | 仮想化支援機能の使用有無 | |
|  |  | EX\_VIRTUALINFO\_DSB:仮想化支援機能を使用しない(デフォルト) | |
|  |  | EX\_VIRTUALINFO\_ENB:仮想化支援機能を使用する | |
|  |  |  | |
|  | <EX\_UNIT\_FLCCROPT> (num=1) | | |
|  | unit\_info[0] | フラッシュメモリ書き換え、消去時にクロック設定変更の可否を指定する。 | |
|  |  | bit0 | 0：EXECから一時的なクロック設定の変更を許可する。  本設定はPhase2.0以前と同等の処理。  　(デフォルト)  1：EXECから一時的なクロック設定の変更を許可しない。 |
|  |  | bit[31:1] | 未使用(0固定) |
|  |  | | |
|  | <EX\_UNIT\_ASYNCPEOPT> (num=2) | | |
|  | unit\_info[0] | 同期/非同期デバッグの選択 | |
|  |  | bit0 | 0：同期デバッグ(デフォルト)  1：非同期デバッグ(1コアデバッグ) |
|  |  | bit[31:1] | 未使用(0固定) |
|  | unit\_info[1] | 非同期デバッグ選択時のデバッグ対象コアの指定 | |
|  |  | 【RH850 G3世代MCUの場合】 | |
|  |  | bit0 | 0：PE1はデバッグ対象外  1：PE1はデバッグ対象 |
|  |  | bit1 | 0：PE2はデバッグ対象外  1：PE2はデバッグ対象 |
|  |  | bit2 | 0：PE3はデバッグ対象外  1：PE3はデバッグ対象 |
|  |  | bit3 | 0：PE4はデバッグ対象外  1：PE4はデバッグ対象 |
|  |  | bit4 | 0：PE5はデバッグ対象外  1：PE5はデバッグ対象 |
|  |  | bit5 | 0：PE6はデバッグ対象外  1：PE6はデバッグ対象 |
|  |  | bit[31:6] | 未使用(0固定) |
|  |  |  |  |
|  |  | 【RH850 G4世代MCUの場合】 | |
|  |  | bit0 | 0：PE0はデバッグ対象外  1：PE0はデバッグ対象 |
|  |  | bit1 | 0：PE1はデバッグ対象外  1：PE1はデバッグ対象 |
|  |  | bit2 | 0：PE2はデバッグ対象外  1：PE2はデバッグ対象 |
|  |  | bit3 | 0：PE3はデバッグ対象外  1：PE3はデバッグ対象 |
|  |  | bit4 | 0：PE4はデバッグ対象外  1：PE4はデバッグ対象 |
|  |  | bit5 | 0：PE5はデバッグ対象外  1：PE5はデバッグ対象 |
|  |  | bit6 | 0：PE6はデバッグ対象外  1：PE6はデバッグ対象 |
|  |  | bit7 | 0：PE7はデバッグ対象外  1：PE7はデバッグ対象 |
|  |  | bit[31:8] | 未使用(0固定) |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <EX\_UNIT\_TRACEOPT> (num=1) | | |
|  | unit\_info[0] | JARL命令不具合の設定 | |
|  |  | bit0 | JARL命令不具合対応の自動設定  0：自動設定で行わない。(将来用。設定しないこと。)  1：自動設定で行う。 |
|  |  | bit1 | bit0=1の時、トレースメッセージ変換機能が使用できる場合の設定。  0：使用する。(本開発バージョンでは設定禁止)  1：使用しない。 |
|  |  | bit[31:2] | 未使用(0固定) |
|  |  |  |  |
|  | <EX\_UNIT\_TRACEMSGPEID> (num=8) | | |
|  | unit\_info[0] | PE0のPEID (参照のみ可能) | |
|  | unit\_info[1] | PE1のPEID (参照のみ可能) | |
|  | unit\_info[2] | PE2のPEID (参照のみ可能) | |
|  | unit\_info[3] | PE3のPEID (参照のみ可能) | |
|  | unit\_info[4] | PE4のPEID (参照のみ可能) | |
|  | unit\_info[5] | PE5のPEID  ex\_rh\_getenv-funcinfo[0]-b5 = 1のとき、かつEX\_UNIT\_SLAVEIDSELでGRM資源が選択されているとき、設定/参照が可能  上記以外：参照のみ可能 | |
|  | unit\_info[6] | PE6のPEID  ex\_rh\_getenv-funcinfo[0]-b5 = 1のとき、かつEX\_UNIT\_SLAVEIDSELでGRM資源が選択されているとき、設定/参照が可能  上記以外：参照のみ可能 | |
|  | unit\_info[7] | PE7のPEID  ex\_rh\_getenv-funcinfo[0]-b5 = 1のとき、かつEX\_UNIT\_SLAVEIDSELでGRM資源が選択されているとき、設定/参照が可能  上記以外：参照のみ可能 | |
|  |  |  | |
|  | <EX\_UNIT\_WORKRAM> (num=2) | | |
|  | unit\_info[0] | ワークRAM領域の開始アドレス | |
|  | unit\_info[1] | ワークRAM領域のサイズ(バイト単位) (参照のみ可能) | |
|  |  |  | |
|  | <EX\_UNIT\_DBGINFOUT>(num=2) | | |
|  | unit\_info[0] | 機能の有効/無効 | |
|  | unit\_info[1] | 任意のデバッグ情報値(32bit) | |
|  | | | |
|  | <EX\_UNIT\_CLUSTERSEL> (num=1) | | |
|  | unit\_info[0] | 操作対象のクラスタ番号  デフォルト値は、実装されているクラスタの最初のクラスタ番号となる。ただし、クラスタが実装されていないデバイスの場合は、不定値となる。 | |
|  | | | |
|  | <EX\_UNIT\_EXTTRACERATE> (num=1) | | |
|  | unit\_info[0] | 外部トレースの転送速度  EX\_EXTTRACERATE\_AUTO : 自動(デフォルト値)  EX\_EXTTRACERATE\_01250M : 1.250[Gbps]  EX\_EXTTRACERATE\_02500M : 2.500[Gbps]  EX\_EXTTRACERATE\_03125M : 3.125[Gbps]  EX\_EXTTRACERATE\_05000M : 5.000[Gbps]  EX\_EXTTRACERATE\_06250M : 6.250[Gbps]  上記以外 : 設定不可 | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <EX\_UNIT\_SVR> (num=9) | | |
|  | unit\_info[0] | SVRパラメータ変更の有効/無効 | |
|  | unit\_info[1] | 入力するSVRパラメータ(SVRCFG0 bit31-0) | |
|  | unit\_info[2] | 入力するSVRパラメータ(SVRCFG1 bit31-0) | |
|  | unit\_info[3] | 入力するSVRパラメータ(SVRCFG2 bit31-0) | |
|  | : | : | |
|  | unit\_info[8] | 入力するSVRパラメータ(SVRCFG7 bit31-0) | |
|  | | | |
|  | <EX\_UNIT\_EFRECOVERY> (num=0) | | |
|  | | | |
|  | <EX\_UNIT\_HEGESEL> (num=1) | | |
|  | unit\_info[0] | 仮想化支援機能使用時のデバッグ対象となるホストモード、ゲストモードとブレーク機能オプション、ステップ機能オプション。  デフォルト値は、0xC00001FF。 | |
|  |  | bit0 | 0：ゲストモードでGPID=0をデバッグ対象としない  1：ゲストモードでGPID=0をデバッグ対象とする |
|  |  | bit1 | 0：ゲストモードでGPID=1をデバッグ対象としない  1：ゲストモードでGPID=1をデバッグ対象とする |
|  |  | bit2 | 0：ゲストモードでGPID=2をデバッグ対象としない  1：ゲストモードでGPID=2をデバッグ対象とする |
|  |  | bit3 | 0：ゲストモードでGPID=3をデバッグ対象としない  1：ゲストモードでGPID=3をデバッグ対象とする |
|  |  | bit4 | 0：ゲストモードでGPID=4をデバッグ対象としない  1：ゲストモードでGPID=4をデバッグ対象とする |
|  |  | bit5 | 0：ゲストモードでGPID=5をデバッグ対象としない  1：ゲストモードでGPID=5をデバッグ対象とする |
|  |  | bit6 | 0：ゲストモードでGPID=6をデバッグ対象としない  1：ゲストモードでGPID=6をデバッグ対象とする |
|  |  | bit7 | 0：ゲストモードでGPID=7をデバッグ対象としない  1：ゲストモードでGPID=7をデバッグ対象とする |
|  |  | bit8 | 0：ホストモードをデバッグ対象としない  1：ホストモードをデバッグ対象とする |
|  |  | bit[29:9] | 未使用(0固定) |
|  |  | bit30 | 0：シングルステップ実行時、1物理コアあたり1命令実行する(従来互換)  1：シングルステップ実行時、デバッグ対象外のCPU動作モードでブレークしない |
|  |  | bit31 | 0：デバッグ対象外のモードでブレークした場合、そのままブレークする  1：デバッグ対象外のモードでブレークした場合、デバッグ対象のモードまで遷移させてブレークする |
|  | | | |
|  | <EX\_UNIT\_INITSTOPDBG> (num=1) | | |
|  | unit\_info[0] | 初期停止・スタンバイモードデバッグ環境の設定/参照 | |
|  |  | EX\_INITSTOPDBG\_DIS: 初期停止・スタンバイモードデバッグ環境を使用しない(デフォルト) | |
|  |  | EX\_INITSTOPDBG\_ENB: 初期停止・スタンバイモードデバッグ環境を使用する | |
|  |  |  | |
|  | | | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_NOSUPRT unit\_typeで指定した機能は未サポート

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_UE\_PLLUNLOCK フラッシュ書き込みのためのPLL駆動でロック状態にできなかった

EX\_UE\_FMSRETIRGVALUE フラッシュマクロサービスから不正値が返却された

EX\_UE\_DFLNOSPRT データフラッシュサポート外デバイス

EX\_PE\_DFLRANGE データフラッシュの範囲を超えた指定をした

EX\_SE\_TRGTOFF ターゲットパワーOFF

EX\_DE\_M\_DCUACCESS DCUアクセス異常

EX\_DE\_M\_MONACCFAILED モニタエリアアクセス失敗

EX\_UE\_M\_BRKRSTFAILED ブレークリセット失敗

EX\_DE\_M\_MONEXEFAILED モニタ実行失敗

EX\_DE\_LPDMODEMISMATCH LPDピンモードが異なっている

EX\_DE\_WRITERIDMISMATCH ライタモードでのIDコード解除失敗

EX\_FE\_WRITERBOOTERROR LPDピンモード操作のためのライタモード起動失敗

EX\_FE\_WRITERENTRYERROR ライタモード接続のモードエントリに失敗

EX\_FE\_WRITERCLOCKERROR ライタモード接続のクロック設定に失敗

EX\_FE\_LPDMODEREADERROR LPDピンモードの読み出しに失敗

EX\_FE\_LPDMODEWRITEERROR LPDピンモードの書き込みに失敗

EX\_FE\_LPDCONNECTERROR LPD通信接続失敗

EX\_FE\_LPDWAKEUPERROR LPDウェイクアップコード読出しに失敗

EX\_FE\_LPDPROTECTERROR LPDプロテクト解除に失敗

EX\_FE\_LPDCONNECTRESET LPD接続に失敗時、リセット端子がアクティブになっていた

EX\_PE\_EXCLUSION 設定値が排他でない

EX\_DE\_TRCNOMOUNT トレース機能は実装されていない

EX\_DE\_SLAVENOMOUNT 指定したスレーブ資源がMCUに実装されていないため、設定できない

EX\_SE\_UNLOCKFAILED セキュリティ解除に失敗した

EX\_SE\_SBRKENBLSLFPRGNOCHG ソフトウェアブレーク設定中にセルフプログラミングのモード変更を行った

EX\_SE\_ACC\_RUN\_PEMODE フラッシュ書き換えモード中(P/Eモード中)のため、

フラッシュ資源の読み出し、および、書き込みは実行できない

EX\_SE\_ACC\_FCUFSEL\_FCU FCUファーム領域選択がFCUファームウェア資源を指定しているため、

読み出し、および、書き込みは実行できない

EX\_SE\_ANOTHERPERUN 別のコアがRUN中のため実行できない

EX\_SE\_ASYNCMODE 非同期デバッグモードに設定できない

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_PE\_INVALIDCORESELECT 指定したPE番号は現在、無効のため選択できない

EX\_SE\_NOSPRT\_RAMINITOFF RAM初期化がOFFのため、指定した機能は実行できない

EX\_PE\_SELTRC 選択されているトレース資源に対しては設定できない

EX\_PE\_MAINOSCLESS MainOSCが未入力

EX\_PE\_NOSPRT\_MAINOSCLESS MainOSCの未入力は未サポート

EX\_DE\_FIDCHKFAILED コードフラッシュ用パスワード認証結果が正常でなかった

EX\_FE\_MAUACCESS MAUにアクセスできない

EX\_PE\_WORKRAMRANGE ワークRAM領域がローカルRAMの範囲を超えた指定をした

EX\_SE\_CANTSET\_ASPON 拡張機能が有効なため設定できない

EX\_SE\_SFTTRACEENB 外部ソフトトレース単線出力が有効

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_FE\_EMUUNKNOWN エミュレータで不明なエラーが発生した

EX\_PE\_NOSPRT\_LPD 指定したLPDは未サポート

EX\_DE\_SELCLUSTERNOMOUNT 選択したクラスタ番号のクラスタは実装されていない

EX\_FE\_TRCMEMCARIB トレースメモリのキャリブレーションに失敗した

EX\_FE\_CNCTEXTTRACE エミュレータ-デバイス間の外部トレースの接続が正しくない

EX\_FE\_CNCTEXTTRACEPLL デバイス側の要因により外部トレースの接続に失敗した

EX\_FE\_CNCTEXTTRACECHUP エミュレータ側の要因により外部トレースの接続に失敗した

EX\_SE\_EMUPWR エミュレータの電源が異常

EX\_UE\_NOSPRT\_DEVEXTTRC 外部トレース未サポートのデバイス

EX\_SE\_DEVTRCPWR デバイスのトレース電源が異常

EX\_FE\_USBFAIL\_EXTTRC 外部トレース接続時にUSBの通信に異常が起きた

EX\_PE\_MAPMODENOMATCH 指定マップモードと実際のマップモードが不一致

EX\_DE\_SINGLEMAPONLY シングルマップモードのみのデバイスである。

EX\_FE\_EFRECOVERYFAILED SWASリカバーコマンドの処理に失敗した。

EX\_DE\_NOSPRT\_SELMAPMODE 選択したマップモードはサポートしていない

EX\_SE\_CPUSTOP\_NOSPRT　　　CPUが停止状態であるためコマンド実行ができない

EX\_SE\_CPUSTOP\_FLASH\_WRITEFAILED CPUが停止中であるためフラッシュ領域へライトできない

EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED 　Cyclicモード中であるためフラッシュ領域へライトできない

EX\_PE\_NOSPRT\_INITSTOPDBG　初期停止状態およびスタンバイモードのデバッグをするモードをサポートしていない

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

＜unit\_type=EX\_UNIT\_LPDOPT＞

[引数説明]

unit\_info[0]は、LPD接続の通信モードの指定

unit\_info[1]は、オプションバイトの処理に関する指定

unit\_info[2]は、LPDクロック周波数(4pin)またはボーレート(1pin)の指定

unit\_info[3]は、LPDクロック周波数(4pin)またはボーレート(1pin)の決定値(接続後に参照のみ可能)

unit\_info[4]は、デバッグクロックの決定値(接続後に参照のみ可能)

unit\_info[5]は、実際に行ったオプションバイト処理の参照

unit\_info[6]は、起動に関するオプション指定

unit\_info[7]は、コールドスタートシーケンスの手順指定

unit\_info[8]は、マップモードの指定

unit\_info[9]～[11]は、予備。将来のため0を設定のこと。

[動作説明]

unit\_type=EX\_UNIT\_RSUで行われるCPUとの接続処理に利用するLPD通信の情報設定を行う。

接続が完了している場合の設定(EX\_SET)は無視する。

unit\_info[0]は、0(1pin)と3(4pin)の設定のみ許可する。左記以外のピンモードも存在するが、設定した場合の動作は保証しない。また、デバイスが1pinモードをサポートしていないのに1pinモードを指定した場合はエラー(EX\_PE\_NOSPRT\_LPD)を返す。IE850の場合は設定を無視して4pin固定となる。

unit\_info[1]は、ビット単位でオプションバイトの処理に関する指定を行う。STMSELがないデバイスで bit4を1にしてもエラーにしないで設定を無視する。ホットプラグインによる接続処理中やIE850の場合、R-Car ICUMXAのようにオプションバイトが存在しない場合は、unit\_info[1]の指定を無視する。

unit\_info[2]の指定に関しては、接続の参考値として扱い、エミュレータで設定可能な設定値に近い値で接続する。なお、0を設定した場合はEXECの既定値で接続を行うが、保障できないため通常利用には推奨しない。IE850の場合でも不測の事態を考慮してLPDクロックの変更可能にすることを推奨する。実際に接続に使用された値は、接続後、参照(EX\_REF)でunit\_info[3]に返る。また、1pinの場合、LPDのボーレートとデバッグクロックは値が異なるため、unit\_info[4]にデバッグクロック値を返す。

unit\_info[5]には実際に行ったオプションバイト処理に関する情報が返る。

unit\_info[6]は、起動時の各種オプション指定を行う。bit0を1にすると、RAM初期化を行わないで起動することが可能。ただし、ホットプラグイン起動の場合は、この設定を無視してRAM初期化は行わない。bit1を1にするとCANブートデバッグでの起動となり、ホットプラグイン接続をすることでCANブートデバッグが可能。

unit\_info[7]は、コールドスタートシーケンスの手順を指定する。端子リセットディレイが発生する構成のPiggyBoardを使用する場合は0を指定する。端子リセットディレイが発生しない構成の評価ボードを使用する場合は1を指定する。デフォルト設定は0となり、BFWが行うコールドスタートシーケンスにウェイトを追加して処理を行う。ウェイトを追加した場合、BIST搭載デバイスの通常シーケンスに対してウェイト時間が約60[ms]追加される。この機能は、デバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合のみ有効となる。ただし、ホットプラグイン接続中の場合には、E1エミュレータによる端子リセット制御が行われないため、func:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合であってもunit\_info[7]による指定は無視される。

unit\_info[8]は、シングルマップモードとダブルマップモード、E2x-FCC2コンパチブルモードがあるデバイスにおいて、マップモードを指定する。複数マップモードを持つデバイスかはデバイスのレジスタを参照して判断する。

デバイスの品種によっては、デバッグ通信I/Fおよびデバイス動作モードの選択を、オプションバイトの書き換えを行なわずエミュレータからの外部端子の制御のみで変更することができる。この場合、unit\_info[1:0]の設定は無視してLPD-4Pinモードで接続する。

また、エミュレータ切断時のデバッグ通信I/Fも初期状態(JTAG)に必ず戻る。

[デバッガ関係者向け注意事項]

(1) RH850G3エミュレータ目標仕様書の起動シーケンスを参考にすること。

エミュレータで設定可能な設定値は、CPUのスペックの最高値とは異なるので、ユーザに選択させる上限には注意が必要。(例としてF1Aのスペックは12MHz以下)

(2) ホットプラグイン接続処理中はunit\_info[1]のオプションバイトの処理に関する指定は無視される。どのような設定値であっても、オプションバイト処理を行わない。

尚、無視されるのは接続時のみであり、切断時はホットプラグイン起動をしていたとしても設定の通りにオプションバイトの処理を行う。

(3) unit\_info[6]でRAMの初期化をしないに選択した場合、デバッガからRAMへのリードアクセスによってもECCエラーが発生する可能性があるため注意が必要。起動直後にユーザプログラムで初期化されていないRAMをメモリウィンドウ等で表示するとECCエラーが発生する。

RAMの初期化をOFFにした場合、以下の機能の一部またはすべてが使用できなくなるため注意。詳細は各関数のDescriptionを参照。

・ex\_memwrite

・ex\_swbrkcond

・ex\_socunitinfo2

(4) unit\_type=EX\_UNIT\_CGでunit\_info[1]=00000000h(メインクロックの入力なし)を設定しているときに、下記の設定をした場合はエラー(EX\_PE\_MAINOSCLESS)を返す。

・unit\_info[0]=0h(1pinモード)

(メインクロックの入力がない場合は、LPD 1pinモードに対応しない。)

・unit\_info[1]:bit28=0(オプションバイト処理を行う)

(メインクロックの入力がない場合は、オプションバイト処理は行えない。)

(5) unit\_info[7]で設定するコールドスタートシーケンスは端子リセットディレイ対策が必要な場合に指定する事。また、対策の必要がないデバイスで指定しても端子リセットディレイ対策を行わないだけで、エラーとはならない。この場合、unit\_info[7]の指定は無視される。

端子リセットディレイ対策はデバイスファイルのEFIタグでfunc:0x0101＜エミュレーション機能情報1＞のbit11(RESET\_DELAY)が1の場合に指定できる。この機能はユーザ選択が必要となり、LPD接続方法の選択であるため、デバッガでの実装はEXEC起動前にデバイスファイルのEFIタグでの判定を行う必要がある。

＜unit\_type=EX\_UNIT\_RSU＞

[引数説明]

セキュリティIDコードが128ビットの場合：

unit\_info[0]～[3]をOCDセキュリティIDコード入力用とし、128ビットまで対応する。

unit\_info[4]に解除結果が返される。

セキュリティIDコードが256ビットの場合：

unit\_info[0]～[7]をOCDセキュリティIDコード入力用とし、256ビットまで対応する。

unit\_info[8]に解除結果が返される。

[動作説明]

func=EX\_SETの場合、OCDセキュリティIDコード(OCDID)解除を行う。CPUとの接続とOCDID解除が正常に行われた場合、EX\_NOERRORが返りunit\_info[4](セキュリティIDコードが256ビットの場合はunit\_info[8])に解除結果として1が設定される。OCDID解除に失敗した場合、unit\_info[4](セキュリティIDコードが256ビットの場合はunit\_info[8])に解除結果として0が設定される。

OCDではこのOCDセキュリティIDコード解除で、CPUとの接続処理が行われる。

CPUとの接続処理では、まずLPDモードの参照と変更が行われる。

LPDモードの参照と変更処理のためにまずエミュレータをライタモードで接続する。接続に失敗した場合はエラー(EX\_FE\_WRITERBOOTERROR)を返す。ライタモードでのIDコード解除に失敗した場合エラー(EX\_DE\_WRITERIDMISMATCH)を返す。LPDモード読み出しに失敗した場合エラー(EX\_FE\_LPDMODEREADERROR)を返す。ライタモードでの接続時に、実際に搭載されているクロックと違う値が設定されていた場合エラー(EX\_FE\_WRITERCLOCKERROR)を返す。ライタモード接続時にFLMD端子の制御失敗などでモードエントリ―に失敗した場合エラー(EX\_FE\_WRITERENTRYERROR)を返す。

LPDモードが初期状態(JTAG)だった場合や指定ピンモードと異なる場合、unit\_type=EX\_UNIT\_LPDOPTの指定に従う。接続を中止する設定になっている場合エラー(EX\_DE\_LPDMODEMISMATCH)を返す。接続を続ける設定の場合は指定ピンモードへの書き換え処理を行う。ピンモード書き換えに失敗した場合エラー(EX\_FE\_LPDMODEWRITEERROR)を返す。これらの一連の処理後にライタモード切断とCPUリセットを行い、引き続きLPD接続処理を行う。LPD接続処理に失敗した場合エラー(EX\_FE\_LPDCONNECTERROR)を返す。LPDのウェイクアップコード読出しに失敗した場合エラー(EX\_FE\_LPDWAKEUPERROR)を返す。LPDのプロテクト解除に失敗した場合エラー(EX\_FE\_LPDPROTECTERROR)を返す。LPD接続に失敗時、リセット端子の状態をモニタし、アクティブ状態になっていた場合は、要因が明確に分かるよう別のエラー(EX\_FE\_LPDCONNECTRESET)を返す。OCDIDコードの解除処理まで行けた場合はEX\_NOERRORとなる。

マップモードが複数存在するデバイスの場合、ex\_socunitinfo2 UNIT\_TYPE=EX\_UNIT\_LPDOPTのunit\_info[8]で指定したマップモードとデバイスのマップモードが一致しているかを確認する。不一致の場合はエラー(EX\_PE\_MAPMODENOMATCH)を返す。シングルマップモードしか存在しないデバイスでダブルマップモードをしている場合はエラー(EX\_DE\_SINGLEMAPONLY)を返す。また、対応していないデバイスでE2x-FCC2コンパチブルモードを指定していた場合、エラー応答(EX\_DE\_NOSPRT\_SELMAPMODE)する。

ex\_initexec()でホットプラグイン起動の指定をしていた場合、LPDモードの参照は行わずに指定モードになっているとみなしてLPD通信接続する。この時もしLPD接続処理に失敗した場合はエラー(EX\_FE\_LPDCONNECTERROR)を返す。接続処理中はCPUリセットを入れず、RUN継続(ホットプラグインRUN)状態で接続処理を完了する。尚、接続処理中にSFRレジスタに対して16バイト分のMAUリードが発生する。この期間は一時的にユーザプログラムの動作が停止する。

IE850の起動でもLPD接続のタイミングのため呼び出す必要がある。OCDIDの値はFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF固定とする。ライタモードの処理は行わない。unit\_info[4]の解除結果は常に1が設定される。

デバイスの品種によっては、デバッグ通信I/Fおよびデバイス動作モードの選択を、オプションバイトの書き換えを行なわずエミュレータからの外部端子の制御のみで変更することができるため、ライタモードモードへ遷移させることなくLPD接続処理を行なう。

デバイスの設定によってはMAUによる接続が行えない場合がある。この場合は、エラー応答(EX\_FE\_MAUACCESS)する。

また、OCDセキュリティIDコードの解除まで完了したタイミングでRAMの初期化(全領域を0でフィル)を実施する。初期化をしないままユーザプログラム実行中にMAUを利用したRAMの読み出しを行うと、ECCエラーが発生してしまうため初期化を行う。Phase2.3より、RAM初期化を行うかどうかはオプション化され、unit\_type=EX\_UNIT\_LPDOPTの指定に従う。ただし、ホットプラグイン接続の場合はRAMの書き換えができないため、初期化は行わない。初期化を行うRAMは以下の通り。ユーザプログラム実行中にMAUアクセスを行う領域が対象となる。

接続時に初期化を行う対象のRAM領域

・Local RAM PEn領域

・Local RAM self領域※1

・Global RAM領域(Retention RAM領域も含む)

・Cluster RAM領域

・Primary RAM領域

・Secondary RAM領域

・FCU RAM領域※2

※1 実態のLocal RAM PEn領域を初期化するため、Local RAM self領域の初期化は必要ないが、PFC1BのようにマルチコアにG3Kが使用されているデバイスでは、Local RAM self領域にLocal RAM PEn領域が割り当てられているため、この場合はLocal RAM self領域も初期化を行う。

※2 FCU RAM領域はMAUアクセスの対象ではないが、ECCエラー制御の対象領域外であるため、ECCエラー発生の対策として初期化を行う。

func=EX\_REFの場合、OCDセキュリティIDコード解除状態の参照を行う。

unit\_info[4](セキュリティIDコードが256ビットの場合はunit\_info[8])に解除結果が設定され、unit\_info[0]～[3](セキュリティIDコードが256ビットの場合はunit\_info[0]～[7])には00000000hが設定される。

[デバッガ関係者向け注意事項]

(1)事前に、unit\_type=EX\_UNIT\_CGでクロック情報の設定およびunit\_type=EX\_UNIT\_LPDOPTでLPD通信の情報設定をしておく必要がある。RH850G3/G4エミュレータ目標仕様書の起動シーケンスを参考にすること。セキュリティIDコード長は、デバイスによって異なるためDFファイル内のEFIタグ「エミュレーション機能情報1 bit21(OCD-ID-WIDTH)」で判断する。

(2)IE850Aエミュレータ接続時、エミュレーションアダプタを使用し電源供給機能を有効に設定した場合は、電源供給を有効に設定したタイミングではなく、本I/Fを実行したときに電源供給を開始する。もし、電源供給に失敗した場合は、本I/Fがエラー(EX\_SE\_EMUPWR)を応答する。なお、エミュレーションアダプタの使用有無の設定はex\_boardinfo2で、電源供給機能の有効/無効の設定はex\_extpowerもしくはex\_initexecで設定する。

＜unit\_type=EX\_UNIT\_CG＞

[引数説明]

デバイスに対する入力クロック情報を指定/参照する。

func=EX\_SET：num=00000004h、unit\_info[]にクロック情報を指定する。

unit\_info[0]：固定値(00000000h)を指定する。

unit\_info[1]：メインクロック(EXTAL/XTAL)に入力している周波数(単位：kHz)を指定する。

入力していない場合は0000000hを指定する。

ただし、メインクロックの入力が必須なデバイスの場合は、00000000hを指定するとエラー(EX\_PE\_NOSPRT\_MAINOSCLESS)を返す。

unit\_info[2]：サブクロックに入力している周波数(単位：Hz)を指定する。なお、固定値(0000000h)を指定すること。

unit\_info[3]：ユーザが設定するPLL逓倍数を指定する。

func=EX\_REF：num=00000004hを返す。

unit\_info[0]：固定値(00000000h)を返す。

unit\_info[1]：func=EX\_SETで指定されたメインクロック(EXTAL/XTAL)に入力している周波数(単位：KHz)を返す。

unit\_info[2]：func=EX\_SETで指定されたサブクロックに入力している周波数(単位：Hz)を返す。固定値(0000000h)を返す。

unit\_info[3]：func=EX\_SETで指定されたユーザが設定するPLL逓倍数を返す。

OCDのホットプラグインRUN中では、エラーにしないで操作可能とする。

＜unit\_type=EX\_UNIT\_FLFUNCINFO＞

[動作説明]

コードフラッシュ関連機能の参照を行う。

unit\_info[0]は、コードフラッシュの領域分割対応情報を返す。

unit\_info[1]は、コードフラッシュの領域数情報を返す。

unit\_info[2]は、FLMD端子の切り替え情報を返す。

FLMD0端子をOCDエミュレータから操作できない場合は、周辺IOレジスタで内蔵プルアップ操作を試みる。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。ただし、FLMD0端子のテストは行わない。

注: 本開発バージョンではFLMD端子の切り替え情報はサポートしない。将来的にサポートされる際にもホットプラグインRUN中はFLMD0端子のテストを行わない。

unit\_info[3]は、フラッシュマクロ種別が返す。サポートしている種別は下記の通り。

FL\_RV40F\_PFC1A RV40FPFC1A時点マクロ

FL\_RV40F\_STD RV40F標準マクロ

FL\_RV28F\_STD RV28F標準マクロ

[デバッガ関係者向け注意事項]

unit\_info[1]は、ROM Less品の場合でも1となる。

判定方法は、unit\_type=EX\_UNIT\_FLMAPINFOを参照すること。

＜unit\_type=EX\_UNIT\_FLMAPINFO＞

[動作説明]

コードフラッシュの先頭アドレスとブロック数を返す。

コードフラッシュが領域分割されている場合は、先頭アドレスとブロック数は以下の様な構成になる。

unit\_info[0] ～ [n-1]は、領域1 ～ 領域nの先頭アドレスを返す。

unit\_info[n] ～ [2n-1]は、領域1 ～ 領域nのブロック数を返す。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

[デバッガ関係者向け注意事項]

ROM Less品の場合、ブロック数でコードフラッシュの有無を判定すること(0ならROM Less品となる)

[コードフラッシュ(ユーザマット)のDualBankについて]

DualBankに関連するunit\_typeが以下の様になる

例：コードフラッシュがDualBankユーザマットとユーザブートマットの構成の場合

・unit\_type=EX\_UNIT\_FLFUNCINFO

unit\_info[1] = 3

・unit\_type=EX\_UNIT\_FLMAPINFO

unit\_info[0] = ユーザマット(Bnak0)の先頭アドレス

unit\_info[1] = ユーザマット(Bnak1)の先頭アドレス

unit\_info[2] = ユーザブートマットの先頭アドレス

unit\_info[3] = ユーザマット(Bnak0)のブロック数

unit\_info[4] = ユーザマット(Bnak1)のブロック数

unit\_info[5] = ユーザブートマットのブロック数

|  |
| --- |
|  |
|  |
| ユーザブートマット |
|  |
|  |
|  |
|  |
|  |
|  |
| ユーザマット |
|  |

[コードフラッシュのマッピングとブロック構成例(ユーザマット:4MBの場合)]

通常マッピング

DualBankマッピング

|  |
| --- |
|  |
|  |
| ユーザブートマット |
|  |
|  |
|  |
|  |
|  |
|  |
| ユーザマット(Bank1)  ユーザマット(Bank0) |
|  |

ブロック数=1

0x01008000

0x01008000

0x01000000

ブロック数=1

0x01000000

0x00A00000

ブロック数=64

0x00800000

0x00200000

ブロック数=70

ブロック数=134

0x00400000

0x00000000

0x00000000

＜unit\_type=EX\_UNIT\_FLBLOCKINFO＞

[動作説明]

コードフラッシュのブロック最終アドレス情報の参照を行う。numにEX\_UNIT\_FLMAPINFOで参照したブロック数以外を指定した場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

[デバッガ関係者向け注意事項]

ブロック情報は、フラッシュメモリの書き込みの際に、上位ソフトがブロック間を跨がないようex\_memwriteを呼び出すために利用するものである。タウンロードでは、できるだけブロックサイズ分のデータを一度に書き込むのが効率良いやりかたである。

＜unit\_type=EX\_UNIT\_DFLNUMINFO＞

[動作説明]

データフラッシュのブロック情報の参照を行う。

unit\_info[0]は、データフラッシュの領域1のブロック数を返す(領域分割対応以前の互換性のため)

unit\_info[1]は、データフラッシュの分割対応情報を返す。また、領域分割対応有の場合は、ブロック数をunit\_type=EX\_UNIT\_DFLMAPINFOで参照する必要がある。

unit\_info[2]は、データフラッシュの領域数を返す。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

[デバッガ関係者向け注意事項]

~~データフラッシュ情報がある場合のブロック番号は、コードフラッシュのブロック番号に続けて割り付ける必要があるので、ex\_socunitinfo2で個別に情報を取得してもブロック番号は連続した番号にする必要がある。デバイスの仕様上の割り付けなのでユーザ見えの表示では注意が必要。~~

＜unit\_type=EX\_UNIT\_DFLMAPINFO＞

[動作説明]

データフラッシュの先頭アドレスとブロック数を返す。

データフラッシュが領域分割されている場合は、先頭アドレスとブロック数は以下の様な構成になる。

unit\_info[0] ～ [n-1]は、領域1 ～ 領域nの先頭アドレスを返す。

unit\_info[n] ～ [2n-1]は、領域1 ～ 領域nのブロック数を返す。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

[データフラッシュのDualBankについて]

DualBankに関連するunit\_typeが以下の様になる

例：データフラッシュがDualBank構成(BankA:64KB、BankB:32KBの2領域)の場合

・unit\_type=EX\_UNIT\_DFLNUMINFO

unit\_info[2] = 2

・unit\_type=EX\_UNIT\_DFLMAPINFO

unit\_info[0] = データフラッシュBnakAの先頭アドレス

unit\_info[1] = データフラッシュBnakBの先頭アドレス

unit\_info[2] = データフラッシュBnakAのブロック数

unit\_info[3] = データフラッシュBnakBのブロック数

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| BankA |
|  |
|  |
|  |

[データフラッシュのマッピングとブロック構成例]

DualBankマッピング

通常マッピング

|  |
| --- |
|  |
| BankB |
|  |
|  |
|  |
|  |
|  |
| BankA |
|  |
|  |
|  |

ブロック数=512

0xFF308000

0xFF300000

0xFF200000

0xFF210000

0xFF210000

ブロック数=1024

ブロック数=1024

0xFF200000

＜unit\_type=EX\_UNIT\_DFLBLOCKINFO＞

[動作説明]

データフラッシュの領域1のブロック最終アドレス情報の参照を行う。領域1はEX\_UNIT\_DFLNUMINFOで参照可能な領域情報の1番目の領域を指す。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

＜unit\_type=EX\_UNIT\_DFLAREABLOCKINFO＞

[動作説明]

データフラッシュの領域別のブロック最終アドレス情報の参照を行う。numにEX\_UNIT\_DFLMAPINFOで参照したブロック数以外を指定した場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

unit\_info[0]には、領域番号n(n=1, 2…)を設定する。

OCDのホットプラグインRUN中では、エラーにしないで参照可能とする。

＜unit\_type=EX\_UNIT\_DCUSEC＞

[引数説明]

unit\_info[0]に設定結果が返る。

unit\_info[1]～[4]を128bitまでのセキュリティID設定に使用する。

[動作説明]

func=EX\_SETの場合、セキュリティIDの設定を行う。

設定時、num=5以外の場合はエラー(EX\_PE\_DATA)を返し、処理は行わない。

セキュリティIDの設定に失敗した場合は、設定結果に0が返り、成功すると1が返る。

セキュリティIDの設定時は設定後にCPUリセットが入る。(厳密にはJTAGのつなげなおしレベルの初期化を行っている)

func=EX\_REFの場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

新セキュリティID設定の前に旧セキュリティIDでのセキュリティ解除処理が行われる。旧セキュリティIDでの解除が失敗した場合エラー(EX\_SE\_UNLOCKFAILED)が返る。

注：本開発バージョンではサポートしない

[デバッガ関係者向け注意事項]

セキュリティIDを設定する機能には、必ず旧セキュリティIDと新セキュリティIDを入力する方法を使うこと。

＜unit\_type=EX\_UNIT\_FLMKOP＞

[引数説明]

unit\_info[0]～unit\_info[7]に32byte分のフラッシュマスクオプションを設定する。

[動作説明]

func=EX\_SETの場合、フラッシュマスクオプションの設定を行う。

設定時、num=8以外の場合はエラー(EX\_PE\_DATA)を返し、処理は行わない。

CPUコアのリセットシーケンスで設定が有効になるため、設定後CPUリセットを実行する。

フラッシュマスクオプションの設定前にセキュリティIDの解除処理が行われる。解除に失敗した場合はエラー(EX\_SE\_UNLOCKFAILED)が返る。

func=EX\_REFの場合、フラッシュマスクオプションの参照を行う。

FCUファーム領域選択レジスタがFCUファームウェア資源を指定している場合は、フラッシュマスクオプションの設定および参照はできないため、エラー(EX\_SE\_ACC\_FCUFSEL\_FCU)を返す。

フラッシュマスクオプションの設定はデバッグモニタ経由でLocalRAM実行により行う。LocalRAM実行処理中に処理が終了しなかった場合、エラー(EX\_DE\_M\_DCUACCESS)を返す。このエラーが返る場合はデバッグモニタが正常に終了できていないため、ex\_resetによって復帰すること。

フラッシュマスクオプションの各bitに割り当てられた機能の中にはエミュレータ接続に関わる重要な機能も割り当てられている。その該当bitを書き換えてしまうとエミュレータとの接続が切れてしまうため、bit単位でマスクして書き換えられないようにする。

なお、PFC1Aはフラッシュマスクオプション機能をサポートせず、EX\_NOSUPRTを返す。

ex\_socunitinfo2のEX\_UNIT\_LPDOPTで「RAM初期化をしない」に設定している場合、エラー(EX\_SE\_NOSPRT\_RAMINITOFF)を返し、処理は行わない。フラッシュマスクオプション書き換え内部処理でRAM領域への書き込みにより、意図せぬ初期化や、ECCエラーが発生してしまうため。フラッシュマスクオプションの設定/参照を行いたい場合は「RAM初期化をする」に設定すること。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

G3世代デバイスでCPU1コアが初期停止状態、またはデバイスがCyclicモード中の場合、Flash資源にアクセスできない。そのため、初期停止状態の場合はエラー応答(EX\_SE\_CPUSTOP\_FLASH\_WRITEFAILED)、Cyclicモード状態の場合はエラー応答(EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED)する。

[デバッガ関係者向け注意事項]

フラッシュマスクオプションは、デバイスシステム仕様でbit割り当てが異なるためデータ自体の設定のみの機能となる。

PFC1Aはフラッシュマスクオプションの設定/参照機能自体をサポートしない。

デバイスのマニュアルには機能が割り当てられたbitが含まれるオプションバイトレジスタOPBTnについての説明しか書いていないが、実際にはOPBT0-7はすべて存在する。EX\_SETで設定する場合、書き換え対象ではないオプションバイトには読み出した値をそのまま設定するようにすること。

＜unit\_type=EX\_UNIT\_FLSECINFO＞

[引数説明]

unit\_info[0]：未使用(0固定)

unit\_info[1]：SecurityFlagの設定/参照

[動作説明]

SecurityFlagとLockBit機能の設定/参照を行う。

func = EX\_SETで、SecurityFlagの設定を行う。設定の反映にはリセットシーケンスが必要になるため、設定後内部的にCPUリセットが実行される。

func = EX\_REFで、SecurityFlagの参照を行う。

unit\_info[1]でSecurityFlagを指定することができる。SecurityFlagの機能はIE850の機能で、E1/E20/E2/IE850Aでは使用できない。E1/E20/E2/IE850Aでは設定値を無視し、参照しても0を返す。

SecurityFlagのbitごとの機能を表 3‑5に示す。SecurityFlagの各bitは0(許可)から1(禁止)に変更可能だが、1(禁止)から0(許可)への変更はできないため、エラー(EX\_PE\_DATA)が返る。一度1(禁止)に変更したbitを0(許可)に戻すにはunit\_type = EX\_UNIT\_FLEXTRAINITでExtraの消去を行う必要がある。

表 3‑5　 フラッシュマクロ種別と設定可能bitの関係

|  |  |  |
| --- | --- | --- |
|  | RV40F |  |
| bit0：ブロック消去禁止フラグ | ○ |  |
| bit1：ライト禁止フラグ | ○ |  |
| bit2：リード禁止フラグ | ○ |  |

○：設定可能

[デバッガ関係者向け注意事項]

SecurityFlag機能はフラッシュセルフ書き込みライブラリで変更されるため、EXEC関数だけで状態が変化する機能ではない。

本開発バージョンでは未サポート。

＜unit\_type=EX\_UNIT\_FLEXTRAINIT＞

[引数説明]

unit\_info[0]：キーコードを入力する。

[動作説明]

func=EX\_SETの場合、コードフラッシュのExtra領域(SecurityFlag、LockBit)を初期化する。設定を反映させるためCPUリセットを実行する。

IE850のみ処理し、E1/E20/E2/IE850Aの場合はなにもしない。

初期化を実行するためには、unit\_info[0]に77777777hのキーコードを設定する必要がある。

func=EX\_REFの場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

[デバッガ関係者向け注意事項]

SecurityFlagの各bitは禁止状態から許可状態に変更できないが、この機能によりSecurtyFlagをすべて許可状態に戻すことができる。

本開発バージョンではSecurityFlagをサポートしないため、EX\_UNIT\_FLEXTRAINITもサポートしない。呼び出してもエラー(EX\_NOSUPORT)が返る。

チップ消去機能はサポートしないが、ex\_memwriteでコードフラッシュ全領域とデータフラッシュ全領域を消去後、ex\_socunitinfo2のunit\_type=EX\_UNIT\_FLEXTRAINITでExtra領域を初期化することで同等の機能を実現可能。

＜unit\_type=EX\_UNIT\_FLLBINFO＞

[引数説明]

func = EX\_SET

unit\_info[0]：設定するLockBit情報

unit\_info[1]：設定する開始ブロック番号を指定

unit\_info[2]：設定する終了ブロック番号を指定

func = EX\_REF

unit\_info[0]：未使用。0固定

unit\_info[1]：参照する開始ブロック番号を指定

unit\_info[2]：参照する終了ブロック番号を指定

unit\_info[3+n]：ブロックnのLockBit情報が返る

[動作説明]

LockBitの設定/参照を行う。ブロックごとのLockBit=0/1の設定/参照を行う。

LockBit=0に設定したブロックの書き込みと消去を禁止できる。ただ、ブレーク中はLockBit機能を内部的に無効にするため、書き込みも消去も可能。

LockBitの設定/参照前にセキュリティIDの解除処理が行われる。解除に失敗した場合はエラー(EX\_SE\_UNLOCKFAILED)が返る。

コードフラッシュがP/Eモード中の場合は、メモリアクセスできないため、エラー(EX\_SE\_ACC\_TUN\_PEMODE)を返す。

FCUファーム領域選択レジスタがFCUファームウェア資源を指定している場合は、LockBitの設定および参照ができないため、エラー(EX\_SE\_ACC\_FCUFSEL\_FCU)を返す。

LockBitの設定/参照はデバッグモニタ経由でLocalRAM実行により行う。LocalRAM実行処理中に処理が終了しなかった場合、エラー(EX\_DE\_M\_DCUACCESS)を返す。このエラーが返る場合はデバッグモニタが正常に終了できていないため、ex\_resetによって復帰すること。

ex\_socunitinfo2のEX\_UNIT\_LPDOPTで「RAM初期化をしない」に設定している場合、エラー(EX\_SE\_NOSPRT\_RAMINITOFF)を返し、処理は行わない。LockBit設定/参照の内部処理でRAM領域への書き込みにより、意図せぬ初期化や、ECCエラーが発生してしまうため。LockBitの設定/参照を行いたい場合は「RAM初期化をする」に設定すること。

ブロック番号はユーザマットの8KBのブロックから32KBのブロックへと連番で規定されている。ユーザブートマットのブロック番号はユーザマットの最終ブロック番号の次の番号とする。DualBankの場合も同様に「ユーザマットのBank0～ユーザマットのBank1～ユーザブートマット」の連番とする。以下にブロック番号の例を示す。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 通常マッピング | | | | DualBankマッピング | | | |
|  | ブロック(n+1) |  | ユーザブート  マット |  | ブロック(m+1) |  | ユーザブート  マット |
|  |  |  |  |
| 0x01000000 |  | 0x01000000 |  |
|  |  |  |  |  |  |  |  |
| 0x00400000 |  |  |  |  |  |
|  | ブロックn |  | ユーザマットの  32KBブロック |  |  |  |
|  |  | 0x00A00000 |  |  |
| 0x003F8000 |  |  | ブロックm |  | ユーザマット  Bank1 |
|  | ： |  |  | ： |  |
|  | ブロック8 |  |  |  |
|  |  |  |  |
| 0x00010000 |  | 0x00800000 | ブロック(n+1) |  |
|  | ブロック7 |  | ユーザマットの  8KBブロック |  |  |  |  |
| 0x0000E000 |  | 0x00200000 |  |  |
|  | ： |  |  | ブロックn |  | ユーザマット  Bank0 |
|  | ブロック1 |  |  | ： |  |
| 0x00002000 |  |  |  |
|  | ブロック0 |  |  |  |
| 0x00000000 |  | 0x00000000 | ブロック0 |  |

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

G3世代デバイスでCPU1コアが初期停止状態、またはデバイスがCyclicモード中の場合、Flash資源にアクセスできないため、エラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

[デバッガ関係者向け注意事項]

コードフラッシュへの書き込み時には、EXEC内部処理でLockBitの設定を一時的に無効にする。このため、EXEC I/F関数経由でのアクセスにはLockBitの設定が影響を及ぼすことはない。

LockBitはコードフラッシュのブロック消去によって該当ブロックのLockBitが初期化されてしまうため、コードフラッシュの消去や、書き込み(消去後に書き込みが行われるため)を行うごとにLockBitが初期化されてしまう。そこで、コードフラッシュの消去や書き込み時に、以下の対策をEXECで行う。

1.消去、書き込み前のLockBitを読み出し、保持する

2.消去、書き込みを実行

3.LockBitが変化したブロックのLockBitを書き戻し

＜unit\_type=EX\_UNIT\_DFLBLNKINFO＞

[引数説明]

unit\_info[0]：ブランクチェック実行の単位を設定する。

unit\_info[1]：ブランクチェック開始ブロック番号/アドレスを設定する。

unit\_info[2]：ブランクチェック終了ブロック番号/アドレスを設定する。

unit\_info[3+0] - [3+m]：ブランクチェック実行結果が格納される。

[動作説明]

データフラッシュのブランクチェックを行い、各ブロックのブランク情報としてブランクチェックの結果をunit\_infoに返す。

func=EX\_REFで使用する。func=EX\_SETの場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

ブロック単位でブランクチェックを行うか、書き込みサイズ単位でブランクチェックを行うかをブランクチェックモード(unit\_info[0])で選択できる。ブロック単位の場合1ブロックから全ブロック数までを指定可能だが、書き込みサイズ単位の場合は16個を上限とする(1ブロックに対して書き込みサイズ単位でブランクチェックすることを想定しているため)。従って、書き込みサイズ単位を指定した場合は、書き込みサイズ単位が16個以下となるよう開始/終了アドレスを指定するよう注意すること。16個を超えるアドレス範囲で指定した場合はエラー(EX\_PE\_DATA)を返す。

ブロック数を超えるブロック番号の指定やデータフラッシュ範囲外のアドレスを指定した場合は、エラー(EX\_PE\_DATA)を返し、処理を行わない。

書き込みサイズ単位でのブランクチェックを行う場合のアドレス指定は、書き込みサイズ境界になっていなくても、内部的に自動で書き込みサイズ境界に変更する。

データフラッシュがP/Eモード中の場合は、メモリアクセスできないため、エラー(EX\_SE\_ACC\_TUN\_PEMODE)を返す。

FCUファーム領域選択レジスタがFCUファームウェア資源を指定している場合は、ブランクチェックが実行できないため、エラー(EX\_SE\_ACC\_FCUFSEL\_FCU)を返す。

データフラッシュのブランクチェックはデバッグモニタ経由でLocalRAM実行により行う。LocalRAM実行処理中に処理が終了しなかった場合、エラー(EX\_DE\_M\_DCUACCESS)を返す。このエラーが返る場合はデバッグモニタが正常に終了できていないため、ex\_resetによって復帰すること。

例)

設定値 実行結果

unit\_info[0] = 1 unit\_info[0] = 1

unit\_info[1] = FF20\_0001h unit\_info[1] = FF20\_0000h (ワードアラインに修正される)

unit\_info[2] = FF20\_0006h unit\_info[2] = FF20\_0007h (ワードアラインに修正される)

unit\_info[3] = 0 or 1 (FF20\_0000h のブランクチェック結果)

unit\_info[4] = 0 or 1 (FF20\_0004h のブランクチェック結果)

ex\_socunitinfo2のEX\_UNIT\_LPDOPTで「RAM初期化をしない」に設定している場合、エラー(EX\_SE\_NOSPRT\_RAMINITOFF)を返し、処理は行わない。データフラッシュブランクチェックの内部処理でRAM領域への書き込みにより、意図せぬ初期化や、ECCエラーが発生してしまうため。データフラッシュブランクチェックを行いたい場合は「RAM初期化をする」に設定すること。

ブロック番号はBank分割なしの場合、ブロック0からブロックnまでの番号となる。DualBankの場合、Bank0に続けてBank1のブロック番号が継続される連番形式とする。以下にブロック番号の例を示す。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bank分割なし | | | | DualBankマッピング | | | |
|  |  |  |  |  | ブロックm |  | Bank1(32KB) |
|  |  | 0xFF307FC0 |  |
|  |  |  | ブロック(m-1) |  |
|  |  | 0xFF307F80 |  |
|  |  |  | ： |  |
|  |  |  | ブロック(n+2) |  |
|  |  | 0xFF300040 |  |
|  |  |  | ブロック(n+1) |  |
|  |  | 0xFF300000 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | ブロックn |  | Bank0(64KB) |  | ブロックn |  | Bank0(64KB) |
| 0xFF20FFC0 |  | 0xFF20FFC0 |  |
|  | ブロック(n-1) |  |  | ブロック(n-1) |  |
| 0xFF20FF80 |  | 0xFF20FF80 |  |
|  | ： |  |  | ： |  |
|  | ブロック1 |  |  | ブロック1 |  |
| 0xFF200040 |  | 0xFF200040 |  |
|  | ブロック0 |  |  | ブロック0 |  |
| 0xFF200000 |  | 0xFF200000 |  |

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

G3世代デバイスではCPU1(G4世代ではCPU0)のコアが初期停止状態、またはデバイスがCyclicモード中の場合、Flash資源にアクセスできないため、エラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

[デバッガ関係者向け注意事項]

データフラッシュのブランクチェック機能は、不定となる読み出しデータの表示制御を行う場合を想定している。ブランクのブロックを表示せず、”?”や”\*”等の記号を表示する場合に使用すること。

ブロック数はunit\_type=EX\_UNIT\_DFLMAPINFOで参照する必要がある。

対象となるのは、ブランク状態では読み出しデータが不定となるデバイスのみとなる。

ブランクチェックモードでブロック単位と書き込みサイズ単位を選択できるが、基本的には最初にブロック単位でブランクチェックを実施し、ブランク以外のブロックに対して書き込みサイズでのブランクチェックを実施することを想定している。

＜unit\_type=EX\_UNIT\_FID＞

[引数説明]

(1)CodeFlash用セキュリティIDコード

unit\_info[0]～[7]をCodeFlashセキュリティIDコード入力用とし、256ビットまで対応する。unit\_info[8]にCodeFlashセキュリティIDコードの認証解除結果を格納する。

CodeFlashセキュリティIDコードの認証が正常に終了しない場合は、エミュレータの起動ができないためエラー(EX\_DE\_FIDCHKFAILED)を返送し、処理を中断する。

(2)DataFlashセキュリティIDコード

unit\_info[9]～[16]をDataFlashセキュリティIDコード入力用とし、256ビットまで対応する。unit\_info[17]にDataFlashセキュリティIDコードの認証解除結果を格納する。

DataFlashセキュリティIDコードの認証が正常に終了しなくても、エミュレータの起動は可能であるため処理を継続する。

なお、G4世代MCUでもセキュリティIDコードの認証が必要であるが、本パラメータ(unit\_type=EX\_UNIT\_FID)とは別のパラメータ(unit\_type=EX\_UNIT\_CMFID)で対応する。

G4世代MCUで本パラメータ(unit\_type=EX\_UNIT\_FID)を発行された場合は、エラー(EX\_NOSUPRT)応答する。

[デバッガ関係者向け注意事項]

(1) 各セキュリティIDコードの認証処理は、本パラメータの発行のタイミングで行わないため、セキュリティIDコードの認証解除結果は、セキュリティIDコード認証処理の後に参照すること。

(2) デバイスのCodeFlashセキュリティIDコードがALL-FまたはReadProtection無効の場合は認証エラーとならず、起動可能。ただし、間違ったIDを入力しているとCodeFlashへのライトはできない。

(3) DataFlashセキュリティIDコードで認証失敗しても起動は可能であるが、DataFlashへのリード/ライトはできない。

(4) デバイスのDataFlashセキュリティIDコードがALL-FまたはReadProtection無効の場合は、間違ったIDを入力しても認証をパスし起動可能だが、DataFlashへのライトはできない。

(5) EX\_UNIT\_LPDOPTで「オプションバイト処理を行う」に設定している場合はCode/DataFlashセキュリティIDがALL-Fかどうか、ReadProtectionが有効かどうかに関わらず、デバイスのセキュリティIDと不一致のIDを入力するとエラー(EX\_DE\_WRITERIDMISMATCH)が返り、接続できない。

各セキュリティIDコード認証が搭載されているMCUの場合は、以下の対応をデバッガで行う。

[CodeFlashセキュリティIDコード認証]

DFファイル内のEFIタグ「エミュレーション機能情報1 bit18(CodeFlash-ID/CustomerID)が“1”の場合は、CodeFlashセキュリティIDコード認証が搭載されているMCUのため、CodeFlashセキュリティIDコードをユーザに入力させる必要がある。また、CodeFlashセキュリティIDコード認証がパスしないとエミュレータは起動できない仕様とする。

[DataFlashセキュリティIDコード認証]

DFファイル内のEFIタグ「エミュレーション機能情報1 bit19(DataFlash-ID)が“1”の場合は、DataFlashセキュリティIDコード認証が搭載されているMCUである。

CodeFlashセキュリティIDコード認証と異なり、DataFlashセキュリティIDコード認証の扱いは以下の通りとする。

・DataFlashセキュリティIDコード認証を入力するためにGUIを用意する。

・DataFlashセキュリティIDコードによる認証を行なわないケースと認証を行うケースをユーザの判断で選ばせる。

DataFlashセキュリティIDコード認証を行なわない場合でも、unit\_info[9]～[16]への値の設定は必須であり、その場合は認証を試みるData FlashセキュリティIDコードとしてALL-Fを設定すること。もし、認証に失敗した場合は、EXEC起動はできるものの、データフラッシュ領域およびプロテクト対象領域へのアクセスが正常に行えないなどの制限が発生する。

補足：DataFlashセキュリティIDコード認証において、ユーザが認証解除を希望しない場合、入力するData FlashセキュリティIDコードをALL-Fとする理由は、RH850 G3世代MCUではExtra領域のセキュリティIDコード格納領域がALL-Fの場合、認証解除の試みを実施する/しないに関わらず、認証が解除されるというデバイス仕様による。RH850では、セキュリティIDコード格納領域がALL-Fの時に認証をしない選択は不可能であるため、認証省略時のダミーセキュリティIDコード値としてALL-Fを入力しても結果に差が発生しないためである。

＜unit\_type=EX\_UNIT\_CMFID＞

[引数説明]

Customer ID/Data Flash IDの認証情報は、以下のようにIDコードごとに指定し、認証するIDコード数分を繰り返し発行する。

num ：11固定

unit\_info[0] ：IDコードタイプ

unit\_info[1] ：IDコードタイプごとのIDコード番号

unit\_info[9:2]：認証するための256ビット分のIDコード

unit\_info[10] ：認証解除結果用(mode=EX\_REF指定時にEXECが格納する)

指定するIDコードタイプとIDコード番号の関係を以下に示す。

|  |  |  |  |
| --- | --- | --- | --- |
| IDコード種別 | IDコードタイプ | IDコード番号 | 備考 |
| Customer ID | EX\_IDTYPE\_CUSTOMER  (0x00000001) | 0x00000001 | n個中の1番目のCustomer IDコードを指定 |
| 0x00000002 | n個中の2番目のCustomer IDコードを指定 |
| 0x00000003 | n個中の3番目のCustomer IDコードを指定 |
| : | ： |
| 0x00000001×n | n個中のn番目のCustomer IDコードを指定 |
| Data Flash ID | EX\_IDTYPE\_DATAFLASH  (0x00000002) | 0x00000001 | m個中の1番目のData Flash IDコードを指定 |
| 0x00000002 | m個中の2番目のData Flash IDコードを指定 |
| 0x00000003 | m個中の3番目のData Flash IDコードを指定 |
| : | ： |
| 0x00000001×m | m個中のn番目のData Flash IDコードを指定 |

存在するCustomer IDコード数(n)およびData Flash IDコード数(m)は、DFファイルのEFI部のCustomer ID認証情報(func=0x0115, index=0x0001)、およびDFファイルのEFI部のData Flash ID認証情報

(func=0x0116, Index=0x0001)から取得して指定すること。

DFファイル上の定義例を以下に示す。

(例1)Customer ID=1個　Data Flash ID=1個の場合

|  |  |  |
| --- | --- | --- |
| DFファイルEFI部 | | 本I/Fの指定例 |
| 情報位置 | 値 |
| Customer ID認証情報  func=0x0115  Index=0x0001 | 0x00000001  (※1) | unit\_info[0]にEX\_IDTYPE\_CUSTOMER  unit\_info[1]に0x00000001  を指定して発行 |
|
|
| Data Flash ID認証情報  func=0x0116  index=0x0001 | 0x00000001  (※1) | unit\_info[0]にEX\_IDTYPE\_DATAFLASH  unit\_info[1]に0x00000001  を指定して発行 |
|
|

※1：Customer ID/Data Flash IDは1個以上存在するため、DFファイル上の情報はそれぞれ必ず0x00000001以上となる。

(例2)Customer ID=3個　Data Flash ID=1個の場合

|  |  |  |
| --- | --- | --- |
| DFファイルEFI部 | | 本I/Fの指定例 |
| 情報位置 | 値 |
| Customer ID認証情報  func=0x0115  index=0x0001 | 0x00000003 | ・unit\_info[0]にEX\_IDTYPE\_CUSTOMER  unit\_info[1]に0x00000001  を指定して発行  ・unit\_info[0]にEX\_IDTYPE\_CUSTOMER  unit\_info[1]に0x00000002  を指定して発行  ・unit\_info[0]にEX\_IDTYPE\_CUSTOMER  unit\_info[1]に0x00000003  を指定して発行 |
| Data Flash ID認証情報  func=0x0116  index=0x0001 | 0x00000001 | unit\_info[0]にEX\_IDTYPE\_DATAFLASH  unit\_info[1]に0x00000001  を指定して発行 |

[動作説明]

起動時に必ず認証が必要なIDコードであるCustomer ID/Data Flash IDを指定する。

Customer ID/Data Flash IDが複数存在する場合は、存在するIDコードごとに、本パラメータを繰り返し発行する。

IDコードの認証処理で、Customer IDに対する本パラメータが1度も発行されていない、または、指定されたCustomer IDのいずれも認証が正常に終了しない場合は、エラー(EX\_DE\_FIDCHKFAILED)を返送して、起動処理を中断する。

なお、いずれか1個の認証が正常に終了し起動処理が完了した場合でも、認証が正常に終了しなかったCustomer IDに対応する領域への命令フェッチを含むアクセスが正常に行えない制限が発生する。

指定されたData Flash IDについては、Data Flash IDに対する本パラメータが1度も発行されていない、または、指定されたすべてのData Flash IDの認証が正常に終了しない場合でもエラー応答はせず、起動処理を継続する。

なお、認証が正常に終了しなかったData Flash IDに対応する領域へのアクセスが正常に行えない制限が発生する。

本パラメータは、G4世代MCUのみサポートするため、G3世代MCU時に指定された場合はエラー(EX\_NOSUPRT)応答する。

[デバッガ関係者向け注意事項]

(1)本パラメータのIDコード設定のための発行は、ex\_socunitinfo2(unit\_type=EX\_UNIT\_RSU)より前に行うこと。

ex\_socunitinfo2(unit\_type=EX\_UNIT\_RSU)より後に発行した場合でもエラーにはしないが、認証処理に反映しない。

(2)各セキュリティIDコードの認証処理は、本パラメータの発行のタイミングで行わないため、セキュリティ認証の解除結果は、セキュリティIDコード認証処理の後に参照すること。

なお、参照した場合、認証するための256ビットのIDコード格納領域にはALL-0を返す。

(3)対象デバイスのCustomer IDコード値がALL-FもしくはALL-0に設定されている場合、デバイスの仕様として無条件で認証が解除される。そのため、認証処理を行なわない、もしくは本パラメータで指定された値がALL-F/ALL-0以外であっても、認証が解除される。

しかし、上記の無条件で認証が解除されるデバイス状態であっても、Customer IDについては、本パラメータをダミーIDコード値としてALL-Fを指定して、必ず1回以上発行すること。

(4)対象デバイスのData Flash IDコード値がALL-FもしくはALL-0に設定されている場合、デバイスの仕様として無条件で認証が解除される。そのため、認証処理を行なわない、もしくは本パラメータで指定された値がALL-F/ALL-0以外であっても、認証が解除される。

Data Flash IDについては、Dalta Flash IDに対する本パラメータが発行されない対応も可能とする。

＜unit\_type=EX\_UNIT\_OPTID＞

[引数説明]

認証情報は、以下のようにIDコードごと指定し、認証するIDコード数分を繰り返し発行する。

num ：10固定

unit\_info[0] ：IDコード番号

unit\_info[8:1]：認証するための256ビット分のIDコード

unit\_info[9] ：認証解除結果用(mode=EX\_REF指定時にEXECが格納する)

指定するIDコード番号の内、以下に示す番号の対象となるIDコードは固定とする。

|  |  |
| --- | --- |
| IDコード番号 | 対象となるIDコード |
| 0x00000001 | Serial ProgrammerID |
| 0x00000002 | C-TEST ID |

存在するIDコード数は、DFファイルのEFI部の任意ID認証情報(func=0x0117, index=0x0001)から取得して指定する。

DFファイル上の定義例を以下に示す。

(例1)ID=2個の場合

|  |  |  |
| --- | --- | --- |
| DFファイルEFI部 | | 備考 |
| 情報位置 | 値 |
| その他のID認証情報  func=0x0117  Index=0x0001 | 0x00000002  (※1) | Serial ProgrammerIDとC-TEST IDの認証のみが可能  ・Serial ProgrammerIDの場合  unit\_info[0]に0x00000001を指定して発行  ・C-TEST IDの場合  unit\_info[0]に0x00000002を指定して発行 |
|
|
|
|

※1：Serial ProgrammerIDとC-TEST IDについてはG4世代共通で必ず存在するため、DFファイル上の情報は必ず0x00000002以上となる。

(例2)ID=4個の場合

|  |  |  |
| --- | --- | --- |
| DFファイルEFI部 | | 本I/Fの指定例 |
| 情報位置 | 値 |
| その他のID認証情報  func=0x0117  index=0x0001 | 0x00000004  (※2) | Serial ProgrammerIDとC-TEST IDのほか2個のIDの認証が可能  ・Serial ProgrammerIDの場合  unit\_info[0]に0x00000001を指定して発行  ・C-TEST IDの場合  unit\_info[0]に0x00000002を指定して発行  ・3個目のIDの場合  unit\_info[0]に0x00000003を指定して発行  ・4個目のIDの場合  unit\_info[0]に0x00000004を指定して発行 |

※2：3個目以降のIDコードはデバイスシリーズごとに異なり、どのIDコードに対応するか、EXECでは把握できない。

[動作説明]

認証が任意であるIDコードを指定する。

複数存在する場合は、存在するIDコードごとに、本I/Fを繰り返し発行する。

IDコードの認証処理で、本I/Fが1度も発行されていない、または、指定されたすべてのIDコードの認証が正常に終了しない場合でもエラー応答はせず、起動処理を継続する。

なお、認証が正常に終了しなかったIDコードに対応する領域へのアクセスが正常に行えない制限が発生する。

本I/Fは、G4世代MCUのみサポートするため、G3世代MCU時に指定された場合はエラー(EX\_NOSUPRT)応答する。

[デバッガ関係者向け注意事項]

(1)本I/FのIDコード設定のための発行は、ex\_socunitinfo(unit\_type=EX\_UNIT\_RSU)より前に行うこと。

ex\_socunitinfo(unit\_type=EX\_UNIT\_RSU)より後に発行した場合でもエラーにはしないが、認証処理に反映しない。

(2)各セキュリティIDコードの認証処理は、本I/Fの発行のタイミングで行わないため、セキュリティ認証の解除結果は、セキュリティIDコード認証処理の後に参照すること。

なお、参照した場合、認証するための256ビットのIDコード格納領域にはALL-0を返す。

(3)本I/Fで指定するIDコードの認証は、エミュレータのデバッグ機能に必須ではないため、存在するIDコードをユーザに通知して、ユーザが認証する/認証しないを選択できるようにする。

(4)対象デバイスのSerial Programmer IDコード値がALL-0に設定されている場合、デバイスの仕様として無条件で認証が解除される。そのため、認証処理を行なわない、もしくは本I/Fで指定された値がALL-0以外であっても、認証が解除される。

(5)対象デバイスのC-TEST IDコード値がALL-FもしくはALL-0に設定されている場合、デバイスの仕様として無条件で認証が解除される。そのため、認証処理を行なわない、もしくは本I/Fで指定された値がALL-F/ALL-0以外であっても、認証が解除される。

＜unit\_type=EX\_UNIT\_COREPESEL＞

RH850G4の場合は、仮想化支援機能の有効/無効にかかわらず、numは、1固定。

[仮想化支援機能を使用しない場合]

[引数説明]

numは、1固定。1以外を指定した場合は、エラー(EX\_PE\_DATA)とする。

func=EX\_SETの場合

unit\_info[0]は、対象PEの設定を行う。

func=EX\_REFの場合

unit\_info[0]は、対象PEの番号を返す。

[動作説明]

個別リソース操作対象の物理CPU(PE)番号を設定する。

デバイスファイルのG3COREIタグに設定されたPE番号であってもPEがDisableとなり選択できない場合がある。この場合、エラー(EX\_PE\_INVALIDCORESELECT)を返しPE切り替えを行わない。

CANブートデバッグ時はPE0以外を選択できず、PE0以外を選択した場合エラー(EX\_PE\_INVALIDCORESELECT)を返す。

ユーザプログラム実行中は、エラーにしないで操作可能とする。

[デバッガ関係者向け注意事項]

(1)デバイスに搭載されているPEの個数および番号については、デバイスファイルを参照する必要がある。エラー(EX\_PE\_INVALIDCORESELECT)が返った場合、指定したPEはDisableであるため使用できないPEであると判定すること。

(2)選択したPE番号が対象とするEXEC I/F、全PEで共通となるEXEC I/Fの情報を以下に示す。

|  |  |
| --- | --- |
| 関数名 | 備考 |
| ex\_reset | 現在選択されているPE番号に関係なく同期リセットのみサポート |
| ex\_regread | 現在選択されているPE番号が対象 |
| ex\_regwrite | 現在選択されているPE番号が対象 |
| ex\_memread | PE番号の選択に関係なく全PEで共通 |
| ex\_memwrite | PE番号の選択に関係なく全PEで共通 |
| ex\_memfill | PE番号の選択に関係なく全PEで共通 |
| ex\_memcopy | PE番号の選択に関係なく全PEで共通 |
| ex\_rh\_evncond | TEU\_CPUイベント条件、CPUコアイベント条件および<TEU\_TCM/TEU\_LRM>イベント条件は、現在選択されているPE番号が対象(TEU\_TCMは未サポート) ※1  <TEU\_GRM/TEU\_CRM>, <TEU\_GIO>, <TEU\_GVC/TEU\_KVC/TEU\_AXI/TEU\_DMA>, <TEU\_EMS>イベント条件は、PE番号の選択に関係なく全PEで共通(TEU\_GIOは未サポート) ※2  各PEの連携イベントは未サポート  イベント条件およびコアイベント条件によるブレークは、リレーブレーク機能を使用した同期ブレークのみサポート |
| ex\_hwbrkevent | TEU\_CPUイベントブレークは、現在選択されているPE番号が対象  ただし、func=EX\_BRK\_REFの場合は、PE番号の選択に関係なく全PEで共通 |
| ex\_swbrkcond | PE番号の選択に関係なく全PEで共通(設定は各PEに同時に行う) |
| ex\_run | 開始アドレスおよびテンポラリブレークの指定は、現在選択されているPE番号が対象  実行は、同期実行のみサポート |
| ex\_step | 現在選択されているPE番号が対象 |
| ex\_stop | PE番号の選択に関係なく同期強制ブレークのみサポート |
| ex\_trcsw | <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>は、現在選択されているPE番号が対象(TCU\_TCMは未サポート) ※1  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>は、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2 |
| ex\_trcrun | PE番号の選択に関係なく全PEで共通 |
| ex\_trcmode | <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>に対するtmodeパラメータは、現在選択されているPE番号が対象(TCU\_TCMは未サポート) ※1  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>に対するtmodeパラメータは、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2  tmode以外のパラメータは、PE番号の選択に関係なく全PEで共通 |
| ex\_trcclear | PE番号の選択に関係なく全PEで共通 |
| ex\_trcevent | <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>は、現在選択されているPE番号が対象(TCU\_TCMは未サポート) ※1  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>は、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2 |
| ex\_trcdelay | PE番号の選択に関係なく全PEで共通 |
| ex\_rh\_trceventsel2 | <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>は、現在選択されているPE番号が対象(TCU\_TCMは未サポート) ※1  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>は、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2 |
| ex\_rh\_trcbufcont | ・trc\_buftypeパラメータにEX\_TRCLTB\_CPUTCMBUFFを指定した場合  現在選択されているPE番号が対象  ・trc\_buftypeパラメータにEX\_TRCLTB\_SLAVE, EX\_TRCLTB\_CLUSTERを指定した場合  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>は、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2  ・trc\_buftypeパラメータに上記以外を指定した場合  PE番号の選択に関係なく全PEで共通 |
| ex\_trcrawmeminfo | PE番号の選択に関係なく全PEで共通 |
| ex\_trcrawmemread | PE番号の選択に関係なく全PEで共通 |
| ex\_rh\_trctune | ・trc\_tunetypeパラメータにEX\_TRC\_CPUSYNCMSG、EX\_TRC\_CPUWATCHMSG、EX\_TRC\_WPTSMSGを指定した場合  <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>は、現在選択されているPE番号が対象(TCU\_TCMは未サポート) ※1  <TCU\_GRM/TCU\_CRM>, <TCU\_GIO>, <TCU\_GVC/TCU\_KVC/TCU\_AXI/TCU\_DMA>, <TCU\_EMS>は、PE番号の選択に関係なく全PEで共通(TCU\_GIOは未サポート) ※2  ・trc\_tunetypeパラメータにEX\_TRC\_CPUMSGCHG、EX\_TRC\_CPUDBMSGを指定した場合  <TCU\_CPU>, <TCU\_TCM>, <TCU\_LRM>は、現在選択されているPE番号が対象(TCU\_TCMは未サポート)  ・trc\_tunetypeパラメータに上記以外を指定した場合  PE番号の選択に関係なく全PEで共通 |
|  |  |

※1 unit\_type=EX\_UNIT\_SLAVEIDSELでスレーブID=0を選択したとき。

※2 unit\_type=EX\_UNIT\_SLAVEIDSELでスレーブID=0以外のスレーブ資源を選択したとき。

[仮想化支援機能を使用する場合]

[引数説明]

numは、3固定。3以外を指定した場合は、エラー(EX\_PE\_DATA)とする。

func=EX\_SETの場合

unit\_info[0]は、対象PEの設定を行う。

注：対象PEが仮想化支援機能に対応していない場合は、下記の設定を行うこと。

unit\_info[1]=0xFF, unit\_info[2]=EX\_COREPESEL\_TYPE\_NM

unit\_info[1]は、マシンもしくはスレッド番号を設定する。

RH850向けDCUでサポート可能な番号まで設定可能。

VMの番号は、0-7まで設定可能

HTの番号は、0-31まで設定可能

NMの番号は、0xFF固定

ただし、対象PEが1VM/1HTではVMおよびHTの番号は0のみ設定できる。0以外の番号を設定した場合は、エラー(EX\_PE\_DATA)を返し処理は行わない。

unit\_info[2]は、マシンもしくはスレッドタイプを以下のデファインから1つ設定する。

EX\_COREPESEL\_TYPE\_NM:選択タイプにNMを設定

EX\_COREPESEL\_TYPE\_VM:選択タイプにVMを設定

EX\_COREPESEL\_TYPE\_HT:選択タイプにHTを設定

func=EX\_REFの場合

unit\_info[0]は、対象PEの番号を返す。

unit\_info[1]は、対象マシンもしくはスレッドの番号を返す。

unit\_info[2]は、対象マシンもしくはスレッドのタイプを返す。

[動作説明]

個別リソース操作対象のPE番号、マシンもしくはスレッド番号、マシンもしくはスレッドタイプを設定する。

デバイスファイルのG3COREIタグに設定されたPE番号であってもPEがDisableとなり選択できない場合がある。この場合、エラー(EX\_PE\_INVALIDCORESELECT)を返しPE切り替えを行わない。

ユーザプログラム実行中、ブレーク中の区別なく設定/参照可能。

[デバッガ関係者向け注意事項]

(1)デバイスに搭載されているPEの個数および番号、仮想化支援機能対応可否については、デバイスファイルを参照する必要がある。

エラー(EX\_PE\_INVALIDCORESELECT)が返った場合、指定したPEはDisableであるため、使用できないPEであると判定すること。また、unit\_info[1]、unit\_info[2]の設定値に関わらず優先してエラーと判定する。

(2)選択した設定によりEXEC I/F関数がどの様な動作になるかは、2.6 仮想化支援機能に関する動作仕様を参照すること。

＜unit\_type=EX\_UNIT\_SWBRKFUNCTION＞

[引数説明]

フラッシュメモリエリアに対するソフトウェアブレークタイプの指定可否情報および実際の使用有無を指定/参照する。

func=EX\_SET：num=00000002hを指定する。

unit\_info[0]：ソフトウェアブレークタイプの指定可否情報をビット単位で返す。

unit\_info[1]：ソフトウェアブレークタイプの使用有無をビット単位で指定する。

func=EX\_REF：num=00000002hを返す。

unit\_info[0]：ソフトウェアブレークタイプの指定可否情報をビット単位で返す。

unit\_info[1]：実際に使用可能なソフトウェアブレークタイプ情報を返す。

なお、numの指定が「2」以外の場合、EX\_PE\_DATAを返す。

[動作説明]

使用できるソフトウェアブレークタイプ情報の参照、およびその情報を利用した実際に使用するソフトウェアブレークタイプの指定を行う。

func=EX\_REF時のunit\_info[1]には、指定されたソフトウェアブレークタイプの内、実際に使用できるソフトウェアブレークタイプの情報を返す。

＜unit\_type=EX\_UNIT\_RUNCONTOPT＞

[引数説明]

ex\_run()時におけるブレーク発生時の各種状態保持情報を指定/参照する。

func=EX\_SET：num=00000001hを指定する。

unit\_info[0]：各種状態保持情報を指定する。

func=EX\_REF：num=00000001hを返す。

unit\_info[0]：設定されている各種状態保持情報を返す。

[動作説明]

内部的に一度ブレークさせて再実行する場合などに、ユーザに対しては実行が継続しているように見せる場合に使用する。

なお、すべての機能を完全に保持することはできないため下記に示す注意が必要である。

・フルストップ/フルブレーク指定時に、bit1=0で再実行した場合、以前のトレース・データが上書きされてしまう。

・各ビットに1を指定して実行する場合、一時ブレーク中に関係する機能の設定(イベント、トレース、RUN-BREAKタイマなど)を変更してはいけない。

・EX\_UNIT\_RUNCONTOPTの機能を1にして実行する場合、一時ブレーク中に関係する機能の設定変更(シーケンシャルイベント設定の変更)をしてはいけない。

関係する機能の設定変更後は、EX\_UNIT\_RUNCONTOPTの機能を0にして初期状態から実行することを推奨する。

一時ブレークした後のex\_run発行前に目的機能のビットを1にして、ex\_run発行後に0に戻す。

常に1にしておく使い方は禁止する。

[デバッガ関係者向け注意事項]

(1)実行中は、エラーにしないで操作可能とする。

＜unit\_type=EX\_UNIT\_DMMRRMOPT＞

[動作説明]

~~ex\_memread()：EX\_M\_DMM\_READ、ex\_memwrite()：EX\_M\_DMM\_WRITEのEXECコマンド処理の際にHALTモードの時のHALTを解除するかどうかの設定を行う。HALT解除の設定は一般公開しないことを推奨する。なお、STOPモードに関しては解除するような設定は設けないで、常にモニタタイムアウトエラーを返すことにする。~~

~~ユーザプログラム実行中は、エラーにしないで操作可能とする。~~

現状のEXECでは未サポート

＜unit\_type=EX\_UNIT\_FLPGOPT＞

[動作説明]

unit\_info[0] bit6を0(コードフラッシュのセルフプログラミングを行う)に設定した場合は、コードフラッシュへのソフトウェアブレークを設定しようとするとエラー(EX\_SE\_SLFPRGONSWBRKNOSPRT)が返る。詳細は、ex\_swbrkcondを参照。

なお、コードフラッシュへのソフトウェアブレークが設定済みの状態で、EX\_UNIT\_FLPGOPTによりコードフラッシュのセルフプログラミングを行う設定(bit6を1)に変更しようとした場合、エラー(EX\_SE\_SBRKENBLSLFPRGNOCHG)を返す。

＜unit\_type=EX\_UNIT\_TRCSEL＞

[引数説明]

トレース機能(TCU)に関連する情報を指定/参照する。

func=EX\_SET：num=00000003hを指定する。

unit\_info[0]：未定義。

unit\_info[1]：使用するトレース出力タイプをビット単位で指定する。

unit\_info[2]：未定義。

func=EX\_REF：num=00000003hを返す。

unit\_info[0]：未定義。

unit\_info[1]：設定されているトレース出力タイプの情報をビット単位で返す。

unit\_info[2]：未定義。

[動作説明]

設定されているトレース出力タイプ情報の参照、および使用するトレース出力タイプの指定を行う。

func=EX\_SETにて使用するトレース出力タイプ(unit\_info[1])を指定すること。なお、外部トレースと内蔵トレースは排他機能であるため、共に使用可能であったとしてもどちらか一つしか使用する指定はできない。両方を指定した場合はエラー(EX\_PE\_EXCLUSION)を返す。外部ソフトトレース単線出力は、外部トレース・内蔵トレースと排他機能ではないので、「外部トレース＋外部ソフトトレース単線出力」もしくは、「内蔵トレース＋外部ソフトトレース単線出力」を設定することは可能となる。また、「外部ソフトトレース単線出力」を単独で設定することも可能である。

外部トレース機能がないエミュレータ(E1/E20/E2)が接続されているときに、外部トレースを選択した場合は、エラー(EX\_DE\_TRCNOMOUNT)を返す。また、外部ソフトトレース単線出力機能がないエミュレータ(E1/E20/IE850/IE850A)が接続されているときに、外部ソフトトレース単線出力を有効にした場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

IE850A接続時に外部トレースが使用できない状態で外部トレースを選択した場合は、エラーを返す。外部トレースが選択できない場合と対応するエラーを下記に示す。

~~・起動時にエミュレータの電源供給異常を検出した場合。(EX\_SE\_EMUPWR)~~

・外部トレース接続時にUSBの通信に異常が起きた場合(EX\_FE\_USBFAIL\_EXTTRC)

・起動時にex\_boardinfo2 - trace\_laneにEX\_TRACELANE\_NONEを指定した場合(EX\_FE\_CNCTEXTTRACEPLLもしくはEX\_FE\_CNCTEXTTRACECHUP)

・外部トレースメモリのキャリブレーションに失敗した(EX\_FE\_TRCMEMCARIB)

・ターゲットデバイスが外部トレースに対応していない場合(EX\_UE\_NOSPRT\_DEVEXTTRC)

・エミュレータのレーン数の設定がターゲットデバイスに対応していない場合(EX\_FE\_CNCTEXTTRACEPLLもしくはEX\_FE\_CNCTEXTTRACECHUP)

・エミュレータのレーン数の設定とターゲットボードの状態が不一致の場合(EX\_FE\_CNCTEXTTRACEPLLもしくはEX\_FE\_CNCTEXTTRACECHUP)

・エミュレータとターゲットデバイス間の設定可能なトレース転送速度が見つからなかった場合(EX\_FE\_CNCTEXTTRACEPLLもしくはEX\_FE\_CNCTEXTTRACECHUP)

・デバイスが電源供給の異常を検出した場合(EX\_SE\_DEVTRCPWR)

トレース出力タイプの選択の変更を行った場合は、トレースの各種設定は初期化される。また、トレース・データもクリアするため、トレースを再開するまで読み出せなくなる。ただし、外部ソフトトレース単線出力の有効／無効だけを切り替えたときは、トレースの各種設定が初期化されることはない。

func=EX\_REF時のunit\_info[1]には、使用しているトレース出力タイプの情報を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)外部ソフトトレース単線出力はサポートしていないため、unit\_info[1]のbit2で「外部ソフトトレース単線出力を使用する」を指定した場合、エラー(EX\_PE\_NOSPRT\_INITSTOPDBG)を返す。

(2)初期停止を除くスタンバイモード状態(ICUM有効の場合のみCyclicRunモードを含む)では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[デバッガ関係者向け注意事項]

(1)デバイスによっては、マスタトレース機能(TCU\_CPUなど)やスレーブトレース機能(TCU\_GRM、TCU\_EMS、TCU\_GVCなど)が一つも実装されていない場合があり、この場合にもし使用する設定をした場合は、エラー(EX\_DE\_TRCNOMOUNT)を返す。マスタトレース機能、スレーブトレース機能の使用可否情報は、ex\_rh\_getenv mcuinfo[0]～[2]を参照のこと。

(2)デバイスによっては、ソフトトレース機能(TCU\_SFT)が一つも実装されていない場合があり、この場合にもし使用する設定をした場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。外部ソフトトレース単線出力機能の使用可否情報は、ex\_rh\_getenv mcuinfo[6]を参照のこと。

(3)デバイスによっては、トレース関連機能(TCU\_TRM、TCU\_TRO、TCU\_GTBなど)が実装されていても、CPUトレース機能(TCU\_CPU)、ローカルRAMトレース機能(TCU\_TCM、TCU\_LRM)については、コアごとに実装されていない場合がある。しかし、1つ以上のコアにCPUトレース機能もしくはスレーブトレース資源が実装され、なおかつトレース出力タイプに外部トレースもしくは内蔵トレースが選択されていれば、ex\_trcswでトレース機能有効の指定が可能となる。

(4)実行中に呼び出した場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

(5)不明な原因でエミュレータがエラー応答した場合は、エラー応答(EX\_FE\_EMUUNKNOWN)する。

[非同期デバッグモード時の注意点]

(1)外部ソフトトレース単線出力機能は非同期デバッグモードに対応しない。もし、非同期デバッグモード時に有効に設定した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。この場合は、内蔵トレース・外部トレースの設定値も反映されず、エラー発生前の状態となる。

＜unit\_type= EX\_UNIT\_FLCCROPT＞

[引数説明]

func=EX\_SETの場合

num=00000001hを指定する。

unit\_info[0]：フラッシュメモリ書き換え、消去時にクロック設定変更の可否を指定する。

EXECから一時的なクロック設定の変更を許可する場合　 00000000h

EXECから一時的なクロック設定の変更を許可しない場合　00000001h

func=EX\_REFの場合

num=00000001hを返す。

unit\_info[0]：クロック設定変更可否の設定値を返す。

[動作説明]

クロック変更を許可した場合、以降のフラッシュメモリへ対する書き換えや消去時に、MCUの動作クロックを設定可能な最高周波数へEXECが一時的に変更することにより処理の高速化を図る。

クロック変更を許可しない場合、EXECはユーザが設定した動作クロックから変更を行わない。

[デバッガ関係者向け注意事項]

クロック変更を許可した場合、ダウンロード性能が向上する一方で、周辺クロックもユーザが意図していない周波数へ変化するため、ユーザシステムへ悪影響を与える可能性がある。

クロック変更を許可しない場合、ユーザシステムへの悪影響は与えないが、ダウンロード性能が許可した場合よりも低下する可能性がある。

本設定はそれぞれにメリットとデメリットがあるため、デバッグ環境にあった設定をユーザへ選択頂くこと。

クロック変更の許可および不許可は、起動後も再設定することにより切り替えることが可能。

ただし、ex\_memwriteコマンドでフラッシュ書き換えを行っている最中で切り替えることはできない。切り替える場合はex\_memwriteコマンドのmodeにEX\_M\_FLASH\_END\_WRをOR指定して、フラッシュ書き換えを完了した後に実施すること。

また、フラッシュメモリエリアへ対するソフトブレークコードの埋め込みなど、書き換えのタイミングがコマンド実行時から遅れる場合、クロック変更の有無は実際の書き換えが実行される直前の設定に従うことに注意すること。

＜unit\_type= EX\_UNIT\_SLAVEIDSEL＞

[引数説明]

func=EX\_SETの場合、対象スレーブIDの設定を行う。

func=EX\_REFの場合、対象スレーブIDの番号を返す。

[動作説明]

個別リソース操作対象のスレーブIDを設定する。

指定したスレーブIDがMCUに実装されていない場合は、エラー(EX\_DE\_SLAVENOMOUNT)を返送する。

ユーザプログラム実行中は、エラーにしないで操作可能とする。

[デバッガ関係者向け注意事項]

(1)デバイスに搭載されているスレーブ資源は、ex\_rh\_getenvで確認する。

(2)選択した対象スレーブIDが対象とするEXEC I/F情報を以下に示す。

|  |  |
| --- | --- |
| 関数名 | 備考 |
| ex\_socunitinfo2 | unit\_type = EX\_UNIT\_TRACEMSGPEIDのとき、PEIDを設定/参照する資源を示す情報としてスレーブIDを使用する。 |
| ex\_rh\_evncond | スレーブ用TEUイベントの条件を設定する場合、イベント設定の対象となる資源を示す情報としてスレーブIDを使用する。 |
| ex\_trcsw | スレーブ用TCUのトレースメッセージ出力動作有効/無効を設定する場合、対象となる資源を示す情報としてスレーブIDを使用する。 |
| ex\_trcmode | スレーブ用TCUに対してトレース動作条件を設定する場合、tmodeパラメータ情報を設定する対象となる資源を示す情報としてスレーブIDを使用する。 |
| ex\_trcevent | スレーブ用TCUに対してトレースイベント条件を設定する場合、対象となる資源を示す情報としてスレーブIDを使用する。 |
| ex\_rh\_trceventsel2 | スレーブ用TCUに対して取得するトレース・データの種別を設定する場合、\*itemパラメータ内b0、b1の資源を示す情報としてスレーブIDを使用する。 |
| ex\_rh\_trctune | スレーブ用TCUに対してトレース機能の調整を行う場合、trc\_tunetypeパラメータがEX\_TRC\_CPUSYNCMSG、EX\_TRC\_CPUWATCHMSG、EX\_TRC\_WPTSMSG指定時の資源を示す情報としてスレーブIDを使用する。 |
| ex\_rh\_evnintgcond | スレーブ用シーケンシャルイベントの条件を設定する場合、シーケンシャルイベント設定の対象となる資源を示す情報としてスレーブIDを使用する。 |
| ex\_hwbrkevent | スレーブ用単体イベントやシーケンシャルイベントのブレーク条件を設定する場合、対象となる資源を示す情報としてスレーブIDを使用する。 |

(3)上記(2)のI/FでCPUもしくはLocalRAMの操作対象PE番号を設定する手順を以下に示す。

(手順1)ex\_socunitinfo2内のunit\_typeパラメータでEX\_UNIT\_SLAVEIDSELを指定し、スレーブID情報を0x00000000(CPUの操作対象PE番号指定)もしくは、スレーブIDを0x00000004(LRM:LocalRAMの操作対象PE番号指定)に設定する。

(手順2)ex\_socunitinfo2内のunit\_typeパラメータでEX\_UNIT\_COREPESELを指定し、操作対象PE番号を設定する。

上記(2)のI/Fに関しては、操作対象PE番号とスレーブIDの両方のパラメータを組み合わせて資源を指定する。スレーブIDが0以外の場合は、操作対象PE番号の指定は無効となり、スレーブIDの情報が有効となる。

(4)上記(2)のI/FでClusterRAMの操作対象クラスタ番号を設定する手順を以下に示す。

(手順1)ex\_socunitinfo2内のunit\_typeパラメータでEX\_UNIT\_SLAVEIDSELを指定し、スレーブID情報を0x00000002(CRM:ClusterRAMの操作対象クラスタ番号指定)に設定する。

(手順2)ex\_socunitinfo2内のunit\_typeパラメータでEX\_UNIT\_CLUSTERSELを指定し、操作対象クラスタ番号を設定する。

＜unit\_type=EX\_UNIT\_VIRTUALINFO＞

[引数説明]

numは、1固定。

func=EX\_SETの場合

unit\_info[0]は、仮想化支援機能使用有無を以下のデファインから1つ選択して設定する。

仮想化支援機能を使用する場合 EX\_VIRTUALINFO\_ENB

仮想化支援機能を使用しない場合 EX\_VIRTUALINFO\_DSB

func=EX\_REFの場合

unit\_info[0]は、仮想化支援機能使用有無の設定値を返す。

[動作説明]

仮想化支援機能の使用有無を設定する。仮想化支援機能に対応しないデバイスの場合は、エラー(EX\_NOSUPRT)を返す。

[デバッガ関係者向け注意事項]

unit\_type=EX\_UNIT\_COREPESELを使用する前に使用有無を設定しておく必要がある。

また、仮想化支援機能使用有無の設定を変更した場合、unit\_type=EX\_UNIT\_COREPESELの設定がNMを選択した状態に変更される。

＜unit\_type=EX\_UNIT\_ASYNCPEOPT＞

[引数説明]

numは、2固定。

func=EX\_SETの場合

unit\_info[0]は、同期デバッグ/非同期デバッグを選択する。

unit\_info[1]は、非同期デバッグ時のデバッグ対象コアを指定する。

func=EX\_REFの場合

unit\_info[0]は、同期デバッグ/非同期デバッグの設定値を返す。

unit\_info[1]は、非同期デバッグ時はデバッグ対象コアのPE情報を返す。同期デバッグ時は0を返す。

[動作説明]

シングルコアのデバイスでunit\_info[0]で非同期デバッグを選択すると、エラー(EX\_PE\_DATA)が返る。

マルチコアのデバイスでunit\_info[0]で同期デバッグを選択した場合、従来通り全コア同時実行・同時ブレークを行い、全コアを同期させてデバッグを行う。

マルチコアのデバイスでunit\_info[0]で非同期デバッグを選択した場合、unit\_info[1]で指定したコアに注目したデバッグを行う。複数コアのbitを1に設定した場合、エラー(EX\_PE\_DATA)を返す。また、存在しないPE番号のbitを1に設定した場合もエラー(EX\_PE\_DATA)を返す。同期デバッグモードでRUN中に本関数を呼び出すと、エラー(EX\_SE\_USRPGMRUN)が返る。非同期デバッグモードでも、別のコアがRUN中の状態で本関数を呼び出すと、エラー(EX\_SE\_ANOTHERPERUN)が返る。

E2拡張機能が有効な状態で非同期デバッグモードに設定しようとすると、エラー(EX\_SE\_CANTSET\_ASPON)が返る。

E2拡張機能が無効な状態かつ外部ソフトトレース単線出力機能が有効な状態で非同期デバッグモードに設定しようとすると、エラー(EX\_SE\_SFTTRACEENB)が返る。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)非同期デバッグはサポートしない。そのため、非同期デバッグを指定した場合、エラー(EX\_PE\_NOSPRT\_INITSTOPDBG)を返す。

[デバッガ関係者向け注意事項]

(1)パラメータを設定した場合、トレース・データのクリアを行う。

[非同期デバッグモード時の注意事項]

(1)トレース出力タイプに内蔵トレースを設定しているとき、なおかつトレース・データの記録動作条件がフルブレーク(EX\_FULL\_BRK)に設定されているときは、非同期デバッグを選択できない。選択した場合はエラー(EX\_SE\_ASYNCMODE)を返す。なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

＜unit\_type=EX\_UNIT\_TRACEOPT＞

[引数説明]

numは、1固定。

func=EX\_SETの場合

トレース関連の設定を行う。

unit\_info[0]-b0は、「1：自動設定を行う」固定。

(「0：自動設定を行わない」は、他の設定項目が追加された場合のための将来用)

unit\_info[0]-b1は、b0=1の時のオプション指定。

func=EX\_REFの場合

エラー(EX\_PE\_DATA)を返す。

[動作説明]

JARL命令不具合対応の自動設定を行った場合、EXECはJARL命令不具合対応に最適なレジスタ設定の組み合わせをデバイスおよびPOD(外部トレース時)に対して行う。

ただし、トレースメッセージ変換機能オプションで「1:使用しない」を設定した場合は、トレースメッセージ変換機能が実装されている場合であっても使用しない。(「0:使用する」の場合は、実装されている場合には使用する。実装されていない場合には使用しない。)

なお、自動設定で設定した項目は、下記のEXEC I/Fで個別に設定/参照することが可能である。

ex\_rh\_trctune trc\_tunetype = EX\_TRC\_CPUSYNCMSG – trc\_tunecond[2]

ex\_rh\_trctune trc\_tunetype = EX\_TRC\_CPUMSGCHG – trc\_tunecond[0]

ex\_rh\_trctune trc\_tunetype = EX\_TRC\_PODCONVMSG – trc\_tunecond[0]

[デバッガ関係者向け注意事項]

(1)パラメータを設定した場合、トレース・データのクリアを行う。

(2)実行中に呼び出したときはエラー(EX\_SE\_USRPGMRUN)を返す。

(3)内蔵トレース、外部トレースを切り替えたときは、設定が初期化されるので自動設定は無効になる。トレース出力先を変更後に必要な場合は、自動設定を再度行うこと。

＜unit\_type=EX\_UNIT\_TRACEMSGPEID＞

numは、8固定。

func=EX\_SETの場合

|  |  |  |  |
| --- | --- | --- | --- |
| ex\_rh\_getenv  funcinfo[0]-b5 | ex\_socunitinfo2 – EX\_UNIT\_SLAVEIDの設定 | PE7～PE5  (unit\_info[7]～[5]) | PE0～PE4  (unit\_info[7]～[5]) |
| 0 | - | 呼び出した場合は、エラー(EX\_PE\_SELTRC)を返す | |
| 1 | GRMを選択 | 物理CPUが実装されているPEへの設定値(PEID=0h, 2h, 3h)は有効。(PEID=1hは禁止、設定した場合は、エラー(EX\_PE\_DATA)を返す。)  実装されていないPEへの設定値は無視する。 | 設定値は無視する。 |
| GRM以外を選択 | 呼び出した場合は、エラー(EX\_PE\_SELTRC)を返す | |

func=EX\_REFの場合

物理CPUが実装されているPE　：割り当てられているPEID(0～3h)を返す。

物理CPUが実装されていないPE：FFFFFFFFhを返す。

[動作説明]

CPUマスタから出力されるトレースメッセージに含まれているPEID(0～3h)について、物理CPUに対する割り当てを設定/参照する。

(例) PFC1Bの割り当て例 (PFC1Bの場合は、設定の変更はできない。)

　PE番号　PEID

PE1 1h

PE2 2h (PFC1Bへの実装はなし)

PE3 3h

PE5 3h (PFC1Bへの実装はなし)

※PE3とPE5が同時に実装されているデバイスが存在しないため、PE3=PE5=3hとなっている。

[マルチコアおよびスレーブ対応時の注意点]

(1)現在選択されている物理CPU(PE)または、スレーブ資源に対して設定する。

(2)物理CPU(PE) または、スレーブ資源の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL/EX\_UNIT\_SLAVEIDSEL)で行う。

[デバッガ関係者向け注意事項]

(1)パラメータを設定した場合、トレース・データのクリアを行う。

(2)実行中に設定で呼び出したときはエラー(EX\_SE\_USRPGMRUN)を返す。参照は正常応答する。

(3)PEIDのデフォルト値は、デバイスやトレース資源によって異なる。

＜unit\_type=EX\_UNIT\_WORKRAM＞

[引数説明]

numは、2固定

func=EX\_SETの場合

unit\_info[0]は、ワークRAM領域の開始アドレスを設定する。

unit\_info[1]は、設定値を無視する。

func=EX\_REFの場合

unit\_info[0]は、ワークRAM領域の開始アドレスを返す。

unit\_info[1]は、ワークRAM領域のサイズ(バイト単位)を返す。

[動作説明]

ローカルRAM上ワークRAM領域の開始アドレスを設定、および開始アドレスとサイズを参照する。

設定/参照したアドレスから参照したサイズ分は、ワークRAM領域としてデバッガのファームウェアが使用する。

ローカルRAMとしてはCPU@PE1(G4世代の場合はCPU@PE0)のローカルRAM領域を用いる。CPU@PE1/PE0のローカルRAM領域がないデバイスでは、プライマリローカルRAM領域を用いる。CPU@PE1/PE0のローカルRAM領域もプライマリローカルRAMもないデバイスの場合はリテンションRAM領域を用いる。

設定アドレスはローカルRAMSelf領域も指定可能であり、CPU@PE1/PE0でのローカルRAMSelf領域として扱う。設定時の指定アドレスがローカルRAMSelf領域の場合、参照時にはローカルRAMSelf領域に対応したCPU@PE1/PE0ローカルRAM領域の実アドレスが読み出せる。

開始アドレスの初期値は、CPU@PE1/PE0のローカルRAM領域またはプライマリローカルRAM領域のあるデバイスにおいて、E1/E20/IE850エミュレータでは0xFEDFDA00、E2/IE850AエミュレータではG3世代MCUが0xFEDFF000、G4世代MCUが0xFDE07000である。CPU@PE1/PE0のローカルRAM領域もプライマリローカルRAM領域もないデバイスにおいて、E1/E20/IE850エミュレータでは0xFEE05A00、E2/IE850Aエミュレータでは0xFEE07000である。

サイズは、E1/E20/IE850エミュレータでは9728byte、E2/IE850Aエミュレータでは4096byteである。

指定アドレスを開始アドレスとしたワークRAM領域がCPU@PE1/PE0のローカルRAM領域/プライマリローカルRAM領域/リテンションRAM領域の範囲に収まらない場合はエラー(EX\_PE\_WORKRAMRANGE)を返す。また、CPU@PE1/PE0のローカルRAM領域またはプライマリローカルRAM領域のあるデバイスにおいて、リテンションRAM領域を指定した場合もエラー(EX\_PE\_WORKRAMRANGE)を返す。

アドレス値は4バイト単位の値を指定すること。アドレス値が4バイト単位でなかった場合、アドレス値の下位2bitを0とし、自動的に補正する。

フラッシュライト中にワークRAM領域の開始アドレスを変更した場合、フラッシュライト完了後(EX\_M\_FLASH\_END\_WRをORしたex\_memwrite発行後)の次のフラッシュライトから適用される。

メモリ内容の退避・復帰が行われるため、ユーザプログラムでもワークRAM領域を使用可能だが、ワークRAM領域は以下の用途では使用できない。

・DMAまたはDTS機能の転送元・転送先

・他の外部マスタでの使用

＜unit\_type=EX\_UNIT\_DBGINFOUT＞

[引数説明]

numは、2固定。

func=EX\_SETの場合

unit\_info[0]：機能の有効/無効を指定する。

00000000h：機能を無効にする(初期値)

00000001h：機能を有効にして、unit\_info[1]の値を設定する。

unit\_info[1]：任意のデバッグ情報値(32bit)を指定する。(unit\_info[0]が有効指定時のみ)

func=EX\_REFの場合

unit\_info[0]：EX\_SETで設定した値を返す。

unit\_info[1]：ユーザプログラムから読み出し可能な領域の値を返す。

[動作説明]

unit\_info[0]に有効指定した場合、unit\_info[1]で指定された任意のデバッグ情報値(32bit)を各コアにあるユーザプログラムから読み出し可能な領域に設定する。なお、全てのコアに対して指定された同じ値を設定する。

unit\_info[0]を有効(00000001h)から無効(00000000h)に変更した場合、各コアにあるユーザプログラムから読み出し可能な領域にデバイス初期値(00000000h)を書き込む。

ex\_rh\_getenvのfuncinfo[7]のbit0が‘1’でかつ、全てのコアがブレーク中の場合のみ設定/参照が可能である。

ex\_rh\_getenvのfuncinfo[7]のbit0が‘0’の場合はエラー応答(EX\_NOSUPRT)する。

選択中のコアが実行中、またはホットプラグインRUN中(接続～強制ブレークするまで)の場合はエラー応答(EX\_SE\_USRPGMRUN)する。

非同期デバッグモードで別のコアが実行中もしくは初期停止中の場合もエラー応答(EX\_SE\_ANOTHERPERUN)する。

本機能は、実際の製品プログラムにデバッグ用処理が組み込まれた状態で、そのユーザプログラム上で設定されたデバッグ情報値からエミュレータが接続されていることを認識し、製品用の処理を動作させるかデバッグ用処理を動作させるかを制御できる環境の提供が目的である。

[デバッガ関係者向け注意事項]

【RH850 G3世代】

(1)デバッグ情報を設定する領域はデバッグモジュール(RCU)のDBG\_STUPレジスタで、EXECからは書き込み/読み出しが可能であるが、CPU側(ユーザプログラム側)は、メモリマッピングされているデバッグ用I/O領域から読み出しのみ可能である。

本機能が割り当たっているデバッグ用I/O領域へのアクセスは、各コアのデバッグ専用領域ベースアドレス(モニタ領域ベースアドレス)＋2078h番地に4バイトアクセスで行なう。

各コアのデバッグ専用領域ベースアドレスは、ex\_rh\_getenvのemuinfoで参照可能である。

(2)本関数はデバイスの品種によって使用できない場合があるため、起動時に使用可否情報をex\_rh\_getenvのfuncinfo[7]で取得すること。

(3)実行中にリセット(WDTリセット、ソフトウェアリセット、端子リセットなど)が発生、もしくはブレーク中にex\_resetコマンドを発行、ex\_resetrunでの実行の場合、設定したデバッグ情報値が初期値に戻ってしまう制限がある。

その場合、一旦全コアをユーザプログラム停止中に遷移させてから以下のいずれかの操作を行なうことで、初期値に戻った値を設定したデバッグ情報値に戻すことが可能である。

・本コマンドをunit\_info[0]= 00000001h(有効)指定で発行して再度デバッグ情報値を設定する。

・本コマンドのunit\_info[0]が有効(00000001h)を指定した状態のまま、再度実行・ステップ実行コマンドを発行(マルチコアで非同期デバッグモードの場合は最初のコアに対して実行・ステップ実行コマンドを発行)

【RH850 G4世代】

(1)デバッグ情報を設定する領域はデバッグインタフェースレジスタ0(DBGIFR0)で、EXECおよびCPU側(ユーザプログラム側)から書き込み/読み出しが可能である。

(2)RH850 G3世代と違い、実行中にリセット(WDTリセット、ソフトウェアリセット、端子リセットなど)が発生、もしくはブレーク中にex\_resetコマンドを発行、ex\_resetrunでの実行の場合でも設定したデバッグ情報値は初期化されない。

＜unit\_type=EX\_UNIT\_CLUSTERSEL＞

[引数説明]

numは、1固定。

func=EX\_SETの場合

unit\_info[0]：操作対象のクラスタ番号を指定する。

func=EX\_REFの場合

unit\_info[0]：EX\_SETで設定した値を返す。

[動作説明]

ex\_socunitinfo2 unit\_type=EX\_UNIT\_SLAVEIDSELで0x00000002(CRM：ClusterRAM)を選択したときの、操作対象のクラスタ番号を設定/参照する。

ex\_socunitinfo2 unit\_type=EX\_UNIT\_SLAVEIDSELでCRM以外を選択したときでも、設定することは可能である。この場合は、CRMを選択したときに設定が有効になる。

実装されていないクラスタ番号を設定した場合は、エラー(EX\_DE\_SELCLUSTERNOMOUNT)を返す。

また、クラスタが実装されていないデバイスに対して、設定した場合は設定値に関係なく必ずエラー(EX\_DE\_SELCLUSTERNOMOUNT)を返す。参照した場合は不定値となる。

本I/Fは、G4世代MCUのみサポートするため、G3世代MCU時に指定された場合はエラー(EX\_NOSUPRT)応答する。

[デバッガ関係者向け注意事項]

(1)デバイスにより実装されているクラスタ数は異なるため、設定可能なクラスタ番号はデバイスによって異なる。設定可能なクラスタ番号の情報はDFファイルから取得する必要がある。

(2)選択した操作対象クラスタ番号が対象とするEXEC I/F情報は、<unit\_type=EX\_UNIT\_SLAVEIDSEL>の[デバッガ関係者向け注意事項]を参照。

(3)<unit\_type=EX\_UNIT\_SLAVEIDSEL>と<unit\_type=EX\_UNIT\_CLUSTERSEL>の関係は、<unit\_type=EX\_UNIT\_SLAVEIDSEL>の[デバッガ関係者向け注意事項]を参照。

＜unit\_type=EX\_UNIT\_EXTTRACERATE＞

[引数説明]

numは、1固定。

func=EX\_SETの場合

unit\_info[0]：外部トレースの転送速度を指定する。

func=EX\_REFの場合

unit\_info[0]：現在設定されている外部トレースの転送速度を返す。

[動作説明]

外部トレースの転送速度を設定／参照する。IE850Aエミュレータ接続時専用の設定項目である。

IE850Aエミュレータ接続以外の時に呼び出した場合は、エラー(EX\_DE\_TRCNOMOUNT)を返す。IE850Aエミュレータ接続時であっても、トレース出力が無効に設定されている場合は、エラー(EX\_DE\_TRCDISABLE)を返す。また、外部トレースが選択されていない場合は、エラー(EX\_PE\_SELTRC)を返す。

デバイスによってはトレース機能(TCU)が未実装の場合もあり、この場合はエラー(EX\_DE\_TRCNOMOUNT)を返す。実装情報の詳細は、ex\_rh\_getenvで参照すること。

func=EX\_SETでunit\_info[0]=EX\_EXTTRACERATE\_AUTO(自動)を指定した場合は、EXECが自動的に設定可能な一番速い転送速度を設定する。“自動”が設定されているときに参照した場合は、unit\_info[0]に実際に設定されている転送速度を返す。

デフォルト値はunit\_info[0]=EX\_EXTTRACERATE\_AUTO(自動)となる。ex\_socunitinfo2 <EX\_UNIT\_TRCSEL>で外部トレースを選択されたときにデフォルトである“自動”の設定を行うが、もし接続可能な転送速度が見つからなかった場合は、外部トレースが使用できないと判断して、ex\_socunitinfo2 <EX\_UNIT\_TRCSEL>はエラー応答する。

ターゲットデバイスとエミュレータに設定している外部トレースI/Fのレーン数の組み合わせによっては、設定できない転送速度が存在する。もし、設定できない組み合わせの転送速度を指定した場合は、エラー(EX\_FE\_CNCTEXTTRACEPLLもしくはEX\_FE\_CNCTEXTTRACECHUP)を返す。

エラー応答時のunit\_infoは不定値となる。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合は、トレース・データのクリアを行う。また、unit\_info[0]=EX\_EXTTRACERATE\_AUTO(自動)を指定した場合も、トレース・データのクリアを行う。

(2)トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。なお、トレース出力タイプ(外部トレース、内蔵トレース)を変更したとき、設定値は初期化される。

＜unit\_type=EX\_UNIT\_SVR＞

[引数説明]

　numは9固定。9以外を指定した場合は、エラー(EX\_PE\_DATA)とする。

func=EX\_SETの場合

(1) unit\_info[0]はunit\_info[1]～[8]のSVRパラメータを入力するか否かを指定する。

　　0：無効。現在のSVRCFG0～7の値をSVRに転送する。

　　1：有効。unit\_info[1]～[8]をSVRCFG0～7にライトし、SVRに転送する。

(2) unit\_info[1]～[8]はSVRCFG0～7に設定する32ビットずつのデータをbit31-0の並びで指定する。

unit\_info[0]が1の場合、unit\_info[1]～[8]の値をSVRCFG0～7にライトする。

　func=EX\_REFの場合

エラー(EX\_PE\_DATA)を返す。

[動作説明]

unit\_type=EX\_UNIT\_RSUで行われるSVRパラメータ設定に利用するSVRパラメータの情報設定を行う。

SVRパラメータはOPBT16～23に格納されており、リセット解除時にSVRCFG0～7に展開される。その後、SVRFSETLCレジスタを操作することでDCDCに転送する。

OPBT16～23に格納されている値が正しければ、unit\_info[0]に0を指定することでDCDCへの転送のみ行い、デバッグが可能となる。OPBT16～23に格納されている値が誤っている場合は、unit\_info[0]に1を指定しunit\_info[1]～unit\_info[8]に正しい値を指定することで、SVRCFG0～7に上書きしDCDCへ転送後にデバッグが可能となる。

SVRパラメータ設定はコールドスタートかつユーザプログラムが自走しない起動の場合のみ必要であり、ホットプラグインやユーザプログラムが自走して起動の場合は不要である。

起動後でも呼び出し可能であるが、設定値はEXEC内部変数に格納されたまま使用されない。

[デバッガ関係者向け注意事項]

(1) SVRパラメータの設定はDFのEFIタグのfunc:0x0103<エミュレーション機能情報3>のbit00(SVR)を参照し、必要な場合に呼び出す。ex\_socunitinfo2 unit\_type=EX\_UNIT\_RSUの前に本I/Fを発行しSVRパラメータの設定を行った後、ex\_socunitinfo2 unit\_type=EX\_UNIT\_RSUの処理内でDCDCへ転送する。

＜unit\_type=EX\_UNIT\_EFRECOVERY＞

[引数説明]

numは0固定。0以外を指定した場合は、エラー(EX\_PE\_DATA)とする。

func=EX\_SETの場合、SWASリカバーコマンドを実行する。

func=EX\_REFの場合、エラー(EX\_PE\_DATA)を返す。

[動作説明]

SWASリカバーコマンドを実行する。

SWASリカバーコマンドは、フラッシュエキストラ領域の書き換え途中で電源遮断等によって処理が中断し、Switch領域がDirty状態の場合に、再度更新し復旧するために用いる。

SWASリカバーコマンドの処理に失敗した場合、SWASリカバーコマンド失敗エラー(EX\_FE\_EFRECOVERYFAILED)を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

G4世代デバイスでCPU0コアが初期停止状態、またはデバイスがCyclicモード中の場合、Flash資源にアクセスできないため、エラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

[デバッガ開発者向け注意事項]

(1)本I/Fはユーザがフラッシュエキストラ領域を復旧させるための手段であるため、デバッガの一つの機能としてUIを実装する必要がある。

＜unit\_type=EX\_UNIT\_HEGESEL＞

[引数説明]

numは、1固定。

func=EX\_SETの場合

unit\_info[0]：操作対象のコアのデバッグ対象とするホスト、ゲストモードとブレーク機能オプション、ステップ機能オプションを指定する。

func=EX\_REFの場合

unit\_info[0]：EX\_SETで設定した値を返す。

[動作説明]

本I/Fでデバッグ対象のCPU動作モードを絞ることによって、特定の仮想マシン上で動作するプログラムやハイパーバイザプログラムのデバッグ効率を上げることができる。

本I/Fの設定により、影響を与える動作を以下に示す。

1. レジスタアクセスにおいて、デバッグ対象外のCPU動作モードの状態ではエラー応答し、レジスタアクセスを抑止する。
2. イベント機能は、デバッグ対象のCPU動作モードでのみイベントを検出する。
3. トレース機能、および、外部ソフトトレース単線出力機能はデバッグ対象のCPU動作モードのトレース・データのみを出力する。
4. パフォーマンス機能は、デバッグ対象のCPU動作モードで実行する区間の現象のみを計測する。

ブレーク機能オプションは、ブレーク発生時にデバッグ対象のCPU動作モードに遷移するまで、EXEC内部で実行するかを指定する。対象とするブレークはステップ実行によるブレークとリセットブレークを除くすべてのブレーク。ステップ機能オプションは、ステップ実行時にデバッグ対象のCPU動作モードのみでステップするかを指定する。

EXECはEX\_UNIT\_HEGESELの設定をコアごとに管理する。ユーザはコアごとにデバッグ対象のCPU動作モードを指定できる。各コアのEX\_UNIT\_HEGESELの未呼び出し状態のデフォルト値は0xC00001FFで、すべてのCPU動作モード、オプションが選択されている。EX\_UNIT\_COREPESELによるコア切替えや、EX\_UNIT\_SLAVEIDSELによるスレーブID切替えのように、選択時のコアやスレーブIDがイベントやトレースの設定に影響するわけではなく、EX\_UNIT\_HEGESELによる設定は最後に設定した値が有効となる。

G4世代MCUで仮想化支援機能が有効な場合に、ex\_socunitinfo2 unit\_type=EX\_UNIT\_COREPESELで指定したコアのデバッグ対象とするCPU動作モード、ブレーク機能オプション、ステップ機能オプションを設定/参照する。

本I/Fは、G4世代MCUのみサポートするため、G3世代ではエラー(EX\_NOSUPRT)応答する。G4世代MCUでも、ICUMコアは仮想化支援機能をサポートしないため、ICUMコアを選択した状態で設定しても無視し、参照した場合は不定値を返す。また、G4世代の仮想化支援機能をサポートしないMCUの場合には、設定しても無視し、参照した場合は不定値を返す。

ターゲットデバイスの仮想化支援機能サポート情報は、デバイスファイルのEFIタグでfunc:0x0103＜エミュレータ機能情報3＞のbit1(Virtualization(G4))で判断できる(1:仮想化支援機能をサポート)。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

＜unit\_type=EX\_UNIT\_INITSTOPDBG＞

[引数説明]

numは1固定。1以外を指定した場合は、エラー応答(EX\_PE\_DATA)する。

func=EX\_SETの場合

unit\_info[0]：初期停止・スタンバイモードデバッグ環境の有効/無効を選択する。

func=EX\_REFの場合

unit\_info[0]は、初期停止・スタンバイモードデバッグ環境の設定値を返す。

[動作説明]

初期停止・スタンバイモードデバッグ環境の有効/無効を設定する。

初期停止・スタンバイモードデバッグ環境に対応しないデバイスで有効を指定した場合はエラー応答(EX\_PE\_NOSPRT\_INITSTOPDBG)する。

また、有効と無効以外の引数を指定した場合はエラー応答(EX\_PE\_DATA)する。

設定の変更は、接続処理の中で行ない、接続処理が完了した以降にfunc=EX\_SETを指定して発行した場合は、エラー応答しないが、設定値は無視する。

[デバッガ関係者向け注意事項]

デバイスが初期停止・スタンバイモードデバッグ環境に対応しているかどうかは、デバイスファイルのEFIタグでfunc:0x0103のbit3(Initial Stop Debug)を確認することにより判定する。

EFI部の詳細についてはデバイスファイルの仕様書を確認すること。

表 3‑6 unit\_type別 EXECの状態と関数の呼出し可(○)/不可(×)の関係

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | unit\_type | EMU | BRK | TRC | TIM | COV | HOT | EMU1 | EMU2 | BRK1 | BRK2 | TRC1 | TRC2 |
| 1 | EX\_UNIT\_LPDOPT | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 2 | EX\_UNIT\_RSU | ○ | ○ | ／ | ／ | ／ | ○ | ○ | ○ | ○ | ○ | ／ | ／ |
| 3 | EX\_UNIT\_CG | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 4 | EX\_UNIT\_FLNUMINFO | － | － | － | － | － | － | － | － | － | － | － | － |
| 5 | EX\_UNIT\_FLFUNCINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 6 | EX\_UNIT\_FLMAPINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 7 | EX\_UNIT\_FLBLOCKINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 8 | EX\_UNIT\_DFLNUMINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 9 | EX\_UNIT\_DFLMAPINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 10 | EX\_UNIT\_DFLBLOCKINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 11 | EX\_UNIT\_DFLAREABLOCKINFO | × | ○ | ／ | ／ | ／ | ○ | × | × | ○ | ○ | ／ | ／ |
| 12 | EX\_UNIT\_DCUSEC | － | － | － | － | － | － | － | － | － | － | － | － |
| 13 | EX\_UNIT\_FLMKOP | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 14 | EX\_UNIT\_FLSECINFO | － | － | － | － | － | － | － | － | － | － | － | － |
| 15 | EX\_UNIT\_FLEXTRAINIT | － | － | － | － | － | － | － | － | － | － | － | － |
| 16 | EX\_UNIT\_FLLBINFO | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 17 | EX\_UNIT\_DFLBLNKINFO | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 18 | EX\_UNIT\_FID | ○ | ○ | ／ | ／ | ／ | ○ | ○ | ○ | ○ | ○ | ／ | ／ |
| 19 | EX\_UNIT\_COREPESEL | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 20 | EX\_UNIT\_SWBRKFUNCTION | × | ○ | ／ | ／ | ／ | × | × | × | ○ | ○ | ／ | ／ |
| 21 | EX\_UNIT\_EXTFLINFO | － | － | － | － | － | － | － | － | － | － | － | － |
| 22 | EX\_UNIT\_RUNCONTOPT | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 23 | EX\_UNIT\_DMMRRMOPT | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 24 | EX\_UNIT\_FLPGOPT | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 25 | EX\_UNIT\_TRCSEL | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 26 | EX\_UNIT\_SLAVEIDSEL | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 27 | EX\_UNIT\_VIRTUALINFO | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 28 | EX\_UNIT\_FLCCROPT | × | ○ | ／ | ／ | ／ | × | × | × | ○ | ○ | ／ | ／ |
| 29 | EX\_UNIT\_ASYNCPEOPT | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 30 | EX\_UNIT\_WORKRAM | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 31 | EX\_UNIT\_DBGINFOUT | × | ○ | × | × | ／ | × | × | × | ○ | × | × | × |
| 32 | EX\_UNIT\_CMFID | ○ | ○ | ／ | ／ | ／ | ○ | ○ | ○ | ○ | ○ | ／ | ／ |
| 33 | EX\_UNIT\_OPTID | ○ | ○ | ／ | ／ | ／ | ○ | ○ | ○ | ○ | ○ | ／ | ／ |
| 34 | EX\_UNIT\_CLUSTERSEL | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 35 | EX\_UNIT\_EXTTRACERATE | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 36 | EX\_UNIT\_SVR | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 37 | EX\_UNIT\_EFRECOVERY | × | ○ | ／ | ／ | ／ | × | × | × | ○ | × | ／ | ／ |
| 38 | EX\_UNIT\_HEGESEL | × | ○ | ／ | ／ | ／ | × | × | × | ○ | ○ | ／ | ／ |
| 39 | EX\_UNIT\_INITSTOPDBG | ○ | ○ | ○ | ○ | ／ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

[備考]

表内の「×」が記載された状態で本関数を呼び出すと、EX\_SE\_USRPGMRUN(ユーザプログラム実行中)のエラーを返す。

表内の「－」は未サポート機能を示す。呼び出すとEX\_NOSUPORT(未サポート)が返る。

### メモリ操作

メモリ・データを操作するための関数。

ex\_memread, ex\_memwrite, ex\_memfill, ex\_memcopy, ex\_memsearch, ex\_memcompare

メモリアクセスおよびSFR領域アクセス時に使用する。

SFR領域アクセス時は隣接するSFRに影響を及ぼさない様に注意すること、また、デバッガ側にてリドロウ対策の代替SFRへのアクセスを行う必要がある。

＜メモリ操作関数のデータバッファ使用方法＞

バイト単位のバッファ(\*data\_1)に対して、指定したアクセスサイズに対応した命令でデータの書き込みもしくは読み出しを行う。

V850E3v5アーキテクチャではデータを**リトル・エンディアン**形式で取り扱うため、この形式に従ったアクセスが行われる。

また、指定アドレスがミスアラインアクセスの対象であってもV850E3v5アーキテクチャに従いミスアラインアクセスが行われる。

・対象メモリ状態

+0 +1 +2 +3 +4 +5 +6 +7 …

0x0000 :00 01 02 03 04 05 06 07 …

・アクセス条件

addr\_1:0x00000000

number:0x00000008

は、アクセスサイズごとのカウント対象範囲(○数字は、countを使用する場合の対象番号を表す)

・アクセスサイズごとの\*data\_1の配置

|  |  |  |  |
| --- | --- | --- | --- |
| acc\_size:EX\_DATA\_SIZE\_1 | | | |
|  |  | +0 +1 +2 +3 +4 +5 +6 +7 |  |
|  | \*data\_1: | 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 |  |
|  |  |  | |
|  |  | ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ |  |
|  | | | |
| acc\_size:EX\_DATA\_SIZE\_2 | | | |
|  |  | +0 +1 +2 +3 +4 +5 +6 +7 |  |
|  | \*data\_1: | 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 |  |
|  |  |  |  |
|  |  | ① ② ③ ④ |  |
|  | | | |
| acc\_size:EX\_DATA\_SIZE\_4 | | | |
|  |  | +0 +1 +2 +3 +4 +5 +6 +7 |  |
|  | \*data\_1: | 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 |  |
|  |  |  |  |
|  |  | ① ② |  |
|  | |  |  |
| acc\_size:EX\_DATA\_SIZE\_8 | | | |
|  |  | +0 +1 +2 +3 +4 +5 +6 +7 |  |
|  | \*data\_1: | 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 |  |
|  |  |  |  |
|  |  | ① |  |
|  | |  |  |
| 注:8バイトアクセスの場合、4バイトのデータを2つの汎用レジスタを組み合わせて使用してアクセスする。 | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memread | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

メモリのデータを読み出す

[Format]

int ex\_memread( EXMEM \*red )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*red | <IO> | メモリアクセス情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <I > | 操作モード | ○ |
|  | ULONG | addr\_1 | <I > | 開始アドレス | ○ |
|  | ULONG | addr\_2 | <--> |  | -- |
|  | ULONG | number | <I > | 範囲のバイト数 | ○ |
|  | USHORT | acc\_size | <I > | アクセスサイズ | ○ |
|  | UCHAR | \*data\_1 | < O> | データバッファポインタ | ○ |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <--> |  | -- |
|  | USHORT | verify\_fg | <--> |  | -- |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | < O> | メモリアクセス手段 | ○ |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | modeは下記の2つのカテゴリに分類され、実行状態およびメモリ領域が決められている。 | |
|  |  | |
|  | 1.ユーザプログラム実行中の処理を指定する。 | |
|  | EX\_NONBR\_RD | ユーザプログラム実行中に呼び出された場合、エラー(EX\_SE\_USRPGMRUN)とする。 |
|  | EX\_BREAK\_RD | ユーザプログラム実行中に呼び出された場合、ex\_stop処理してからデータを読み出す。その後プログラムを再実行する。 |
|  | EX\_M\_DMM\_RD | ユーザプログラム実行中に呼び出された場合、MAUを使用してデータを読み出す。 |
|  |  |  |
|  | 2.外付けフラッシュメモリ書き換え時の処理を指定する。 | |
|  | ~~EX\_M\_EXTFLASH\_ID\_RD~~ | ~~外付けフラッシュメモリのID情報を読み出す。~~ |
|  | 本開発バージョンでは、外付けフラッシュをサポートしない | |
|  |  |  |
| addr\_1 | | |
|  | <address value> | 読み出し開始アドレス |
|  |  |  |
| number | | |
|  | <1-00100000h> | 読み出し範囲のバイト数 |
|  |  | ラップラウンドは不可 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| acc\_size | | |
|  | EX\_DATA\_SIZE\_1 | 1バイトアクセス |
|  | EX\_DATA\_SIZE\_2 | 2バイトアクセス |
|  | EX\_DATA\_SIZE\_4 | 4バイトアクセス |
|  | EX\_DATA\_SIZE\_8 | 8バイトアクセス |
|  |  |  |
| \*data\_1 | | |
|  | <0-FFh> | 読み出しデータ |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| direct | | | |
|  | メモリアクセスのアクセス手段情報を格納する。 | | |
|  | 【RH850 G3世代MCUの場合】 | | |
|  | bit0-2 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 000b:  CPU経由でアクセスしていない。  001b～111b:  CPU経由でアクセス時のアクセスしたPE番号。  001bの場合、CPU@PE1のアクセスを示す。 |
|  | bit3-5 | EXECのアクセス手段2  MAU経由でアクセスしたときに使用したMAUのPE番号 | 000b:  MAU経由でアクセスしていない。  001b～111b:  MAU経由でアクセス時のアクセスしたPE番号。  011bの場合、MAU@PE3でのアクセスを示す。 |
|  | bit6 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit7 | 未定義 |  |
|  |  |  |  |
|  | 【RH850 G4世代MCUの場合】 | | |
|  | bit0-3 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 0000b:  CPU経由でアクセスしていない。  0001b～1111b:  CPU経由でアクセス時のアクセスしたPE番号。  0001bの場合、CPU@PE0のアクセスを示す。  ：  1000bの場合、CPU@PE7のアクセスを示す。 |
|  | bit4-7 | EXECのアクセス手段2  MAU経由でアクセスしたときのカレントPE番号 | 0000b:  MAU経由でアクセスしていない。  0001b～1111b:  MAU経由でアクセス時のカレントPE番号。  0001bの場合、PE0選択時にMAUアクセスの発生を示す。  ：  1000bの場合、PE7選択時にMAUアクセスの発生を示す。 |
|  | bit8 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit9-31 | 未定義 |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_FE\_M\_MONTIMOUT\_MEM メモリ関数モニタタイムアウト

~~EX\_UE\_BROMAREA 裏ROMエリアを指定した~~

~~EX\_UE\_EXTFLASHENV 外付けフラッシュメモリのID情報読み出し不可状態~~

EX\_NOSUPRT 本関数は未サポート

EX\_DE\_MAU\_NOSUPRTAREA 指定したエリアは、MAUアクセスをサポートしていないためアクセスできない

EX\_PE\_OVERLAP\_MCURSC 指定したエリアは、MCUの資源を跨ぐためアクセスできない

EX\_SE\_ACC\_RUN\_PEMODE フラッシュ書き換えモード中(P/Eモード中)のため、フラッシュ資源の読み出しおよび、書き込みはできない

EX\_SE\_ACC\_FCUFSEL\_FCU FCUファーム領域選択がFCUファームウェア資源を指定しているため、読み出しおよび、書き込みはできない

EX\_SE\_AXICLKDSB\_ACCFAILED AXIクロックが供給されてないため、XC領域にアクセスできない

EX\_SE\_SFLCLKDSB\_ACCFAILED Serial Flashにクロックが供給されていないため、Serial Flash領域にアクセスできない

EX\_SE\_SDRAMCLKDSB\_ACCFAILED SDRAMにクロックが供給されていないため、SDRAM領域にアクセスできない

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_DE\_SECAREAACCERR　　　　 指定された領域にセキュア領域が含まれるため選択されたコアではアクセスできない

EX\_SE\_CPUSTOP\_AREA\_ACCFAILED　 CPUが停止中であるため指定した領域にアクセスできない

EX\_SE\_STANDBY\_NOSPRT 該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

読み出し開始アドレス(addr\_1)からバイト数(number)バイト分のメモリをdata\_1が指し示す領域にacc\_sizeの単位で読み出す。

[引数説明]

modeは、指定のカテゴリごとに説明が分類される。

＜カテゴリ別の説明＞

1.ユーザプログラム実行中の処理

ブレーク中の場合は、実行状態による処理の相違なく読み出しを行う。

フラッシュメモリ、外付けフラッシュメモリ、もしくはユーザプログラム実行中にアクセスできない領域を指定した場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

~~2.外付けフラッシュメモリ書き換え時の処理~~

~~ユーザプログラム実行中の場合は、エラー(EX\_SE\_USRPGMRUN)を返す。~~

~~外付けフラッシュメモリ以外の領域を指定した場合は、エラー(EX\_PE\_M\_NOEXTFLAREA)を返す。~~

注:本開発バージョンでは、外付けフラッシュをサポートしない

numberが、アクセスサイズで割り切れない場合、余ったデータを無視する。

[デバッガ関係者向け注意事項]

SFR(周辺I/O)アクセス時に使用する場合、リドロウ対策の代替SFRを考慮したアドレスを指定する必要がある。

ECC無効化処理を行うため、デバイス仕様のメモリマッピングで定義されている領域ごとに本関数を呼び出す必要がある。複数の領域を跨ぐ指定を行わないこと。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

ホットプラグインRUN中(接続～強制ブレークするまで)は、mode=EX\_BREAK\_RDをEX\_SE\_USRPGMRUNエラーとする。EX\_M\_DMM\_RDを利用すること。

[動作説明]

mode別の動作を説明する。

|  |  |  |
| --- | --- | --- |
| <EX\_NONBR\_RD> | | |
|  | ブレーク中のみ読み出し処理を行い、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)を返す。 | |
|  | | |
| <EX\_BREAK\_RD> | | |
|  | EXEC関数をex\_stop() → ex\_memread() → ex\_run()の順に呼び出したのと同じ操作をする。  ブレークの後処理を完了した状態でデータを読み込む。 | |
|  | | |
| <EX\_M\_DMM\_RD> | | |
|  | G3世代MCUの場合はLocal RAM領域、Global RAM領域、Retention RAM領域、Primary RAM領域、Secondary RAM領域のみ、G4世代MCUの場合はG3世代MCUの領域に加えてCluster RAM領域、周辺I/O領域をMAUを使用してアクセス可能。MAUはDMAと同様にユーザプログラム動作のWAITを発生させるため、完全にユーザプログラム実行が連続して行われる訳ではない。  上記以外の領域に対して、指定した場合はEX\_DE\_MAU\_NOSUPRTAREAを返す。  [デバッガ関係者向け注意事項]  D1xのRetention RAM領域は対象外(クロスコネクト領域内にあるため)。  周辺I/O領域はユーザによるアクセス制限が設定可能なため本開発バージョンではサポートしない。 | |
|  | | |
| ~~<EX\_M\_EXTFLASH\_ID\_RD>~~ | | |
|  | | ~~ターゲット環境に実装されている外付けフラッシュメモリデバイスに対して、ID情報読み出しコマンドシーケンスを発行(実際はフラッシュメモリに対してID読み出し用コマンドシーケンスのライト処理)した上で、指定されたアドレスから、指定されたバイト数分のデータを読み出し、\*data\_1へ返します。(ID情報読み出し後は自動的にフラッシュメモリデバイスを"通常モード"へ戻すため、後に他のmodeでメモリリードを実行してもID情報は読み出せず、フラッシュメモリデバイスの指定アドレスに該当するメモリ・データが読み出される。)~~  ~~フラッシュメモリのデータを変更することなく、「ID情報が正しく読み出されるかどうか」を確認することで、ターゲット環境、SFR、フラッシュ情報等の設定が正しく行われていることを確認することが可能。~~  ~~パラメータのacc\_sizeは無視し、ex\_socunitinfo2()であらかじめ設定されたデータバス幅に従ってリードアクセスをする。よって、事前のex\_socunitinfo2()による、"フラッシュ情報の設定"手続きがなされていない状態でリードした場合は、エラーであるEX\_UE\_EXTFLASHENV(外付けフラッシュメモリのID情報読み出し不可状態)を返す。~~  ~~また、addr\_1 + numberで指定したリードアドレス範囲がフラッシュエリア外を含む場合はEX\_PE\_DATAを返す。~~ |
| 注:本開発バージョンでは、外付けフラッシュをサポートしない | | |
|  | | |
| E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。 | | |

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

G4世代のFlash領域へのメモリアクセスでは、選択されているコアから指定された領域にアクセスする。

指定された領域にセキュア領域が含まれていてかつ、選択されているコアがセキュア領域にアクセスできない場合、エラー(EX\_DE\_SECAREAACCERR)を返す。

また、セキュリティコアを有効にした動作モード時、ユーザのデバイス設定によっては正しい値が読みだせない場合がある。正しい値を読み出すには、ユーザ自身でデバイスの設定を行なう必要がある。

|  |  |
| --- | --- |
| ~~[外付けフラッシュメモリID読み出し時の注意事項]~~ | |
| ~~・ID情報である「メーカーコード(ベンダーコード)」、「デバイスコード」情報は、フラッシュメモリエリア内のどこのアドレスを指定しても読み出しが可能。(推奨アドレス＝任意のセクタ(ブロック)の先頭アドレス)~~  ~~指定アドレス(addr\_1)値の下1桁(16byte)内で、それぞれの情報の場所がマッピングされており、この位置はフラッシュメモリデバイスの並列実装個数の状況によって変化するため注意が必要。~~ | |
| ~~・但し、フラッシュ情報の設定に不備があり、別のメモリエリアをフラッシュエリアとして設定した場合、ID情報読み出し用コマンドシーケンスのライト処理によって、メモリ上の元のデータをコマンドデータで上書きしてしまうおそれがある。~~ | |
|  | |
| ~~以下に並列実装状況ごとの情報マッピングを示す。(注:SAn = n番目セクタの先頭のアドレス)~~ | |
| ~~<並列実装個数＝ 8bit幅デバイス 1個の場合>~~ | |
|  | ~~+0 +1 +2 +3 +4 +5 +6 +7 … +F~~  ~~SAn：XX YY -- -- -- -- -- -- --~~  ~~XXh = メーカーコード(ベンダーコード)~~  ~~YYh = デバイスコード~~ |
| ~~<並列実装個数＝ 8bit幅デバイス 2個の場合> … 2デバイスそれぞれからID情報が読み出される~~ | |
|  | ~~+0 +1 +2 +3 +4 +5 +6 +7 … +F~~  ~~SAn：XX XX YY YY -- -- -- -- --~~  ~~XXh = メーカーコード(ベンダーコード)~~  ~~YYh = デバイスコード~~ |
| ~~<並列実装個数＝ 8bit幅デバイス 4個の場合> … 4デバイスそれぞれからID情報が読み出される~~ | |
|  | ~~+0 +1 +2 +3 +4 +5 +6 +7 … +F~~  ~~SAn：XX XX XX XX YY YY YY YY --~~  ~~XXh = メーカーコード(ベンダーコード)~~  ~~YYh = デバイスコード~~ |
| ~~<並列実装個数＝16bit幅デバイス 1個の場合>~~ | |
|  | ~~+0 +2 +4 +6 … +E~~  ~~SAn：XXXX YYYY ---- ---- ----~~  ~~XXXXh = メーカーコード(ベンダーコード)~~  ~~YYYYh = デバイスコード~~ |
| ~~<並列実装個数＝16bit幅デバイス 2個の場合> … 2デバイスそれぞれからID情報が読み出される~~ | |
|  | ~~+0 +2 +4 +6 … +E~~  ~~SAn：XXXX XXXX YYYY YYYY ----~~  ~~XXXXh = メーカーコード(ベンダーコード)~~  ~~YYYYh = デバイスコード~~ |
| ~~<並列実装個数＝32bit幅デバイス 1個の場合>~~ | |
|  | ~~+0 +4 … +C~~  ~~SAn：XXXXXXXX YYYYYYYY --------~~  ~~XXXXh = メーカーコード(ベンダーコード)~~  ~~YYYYh = デバイスコード~~ |
| 注:本開発バージョンでは、外付けフラッシュをサポートしない | |

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)操作対象のPEが停止(初期停止・Cyclic Disable)中で、アクセス領域のメモリアクセス手段がCPUアクセスの場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_AREA\_ACCFAILED)する。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

(2)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

|  |
| --- |
| [ECC付メモリ読み出し時の注意事項] |
| G3M、G4MHではECCが確定していない状態でECC付メモリを読み出すと、ECCエラーが発生する。このため、メモリ読み出し時には、一時的にECCを無効にして処理を行う。以下の4種類の領域の場合、ECCの設定を変更する。  ＜ECC設定変更対象領域＞  1.CodeFlash  消去状態ではデータとECCが一致しないため、CodeFlash用ECCを一時的に無効にする。  2.DataFlash  消去状態はDataFlashのデータが不定状態となりECCが確定しないため、DataFlash用ECCを一時的に無効にする。  3.その他  ECCを一時的に無効にする制御対象は、MCU品種に依存する。  詳細は、F/W機能仕様書を参照。 |
| [FCUに関連する注意事項] |
| ・メモリアクセスするフラッシュ資源がP/Eモード中は、メモリアクセスを行わず  EX\_SE\_ACC\_RUN\_PEMODEを返送する。  ・FCUファーム領域選択レジスタがFCUファームウェア資源を指定し、コードフラッシュを対象として  メモリアクセスを行った場合は、メモリアクセスを行わずEX\_SE\_ACC\_FCUFSEL\_FCUを返送する。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memwrite | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

メモリにデータを書き込む

[Format]

int ex\_memwrite( EXMEM \*wrt )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*wrt | <IO> | メモリアクセス情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <I > | 操作モード | ○ |
|  | ULONG | addr\_1 | <I > | 開始アドレス | ○ |
|  | ULONG | addr\_2 | < O> | ベリファイエラーアドレス | ○ |
|  | ULONG | number | <--> |  | -- |
|  | USHORT | acc\_size | <I > | アクセスサイズ | ○ |
|  | UCHAR | \*data\_1 | <I > | データバッファポインタ | ○ |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <I > | 書き込みデータのアクセスサイズ単位の個数 | ○ |
|  | USHORT | verify\_fg | <I > | ベリファイフラグ | ○ |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | < O> | メモリアクセス手段 | ○ |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mode | | | | |
|  | modeは下記の3つのカテゴリに分類され、実行状態およびメモリ領域が決められている。 | | | |
|  |  | | | |
|  | 1.ユーザプログラム実行中の処理を指定する。 | | | |
|  | EX\_NONBR\_WR | | ユーザプログラム実行中に呼び出された場合、エラー(EX\_SE\_USRPGMRUN)とする。 | |
|  | EX\_BREAK\_WR | | ユーザプログラム実行中に呼び出された場合、ex\_stop処理してからデータを書き込む。その後プログラムを再実行する。 | |
|  | EX\_M\_DMM\_WR | | ユーザプログラム実行中に呼び出された場合、MAUを使用してデータを書き込む。 | |
|  |  | | | |
|  | 2.フラッシュメモリ書き換え時の処理を指定する。 | | | |
|  | EX\_M\_FLASH\_PV\_WR | | コードフラッシュ、データフラッシュメモリおよびフラッシュエキストラ領域への書き込みを行う。  注：フラッシュエキストラ領域への書き込みはRH850 G4世代MCUのみサポートする。 | |
|  | EX\_M\_FLASH\_E\_WR | | コードおよびデータフラッシュメモリの消去のみ行う。 | |
|  | EX\_M\_FLASH\_EP\_WR | | コードフラッシュメモリへの書き込みを行う。 | |
|  | EX\_M\_FLASH\_P\_WR | | コードフラッシュメモリへの書き込みのみ行う。 | |
|  | EX\_M\_FLASH\_END\_WR | | フラッシュダウンロードの終了フラグ。上記3つのデファインとORして指定する。  注：フラッシュメモリ書き換えはEX\_M\_FLASH\_END\_WRをORして終了させる必要がある。 | |
|  |  | | | |
|  | 3.外付けフラッシュメモリ書き換え時の処理を指定する。 | | | |
|  | ~~EX\_M\_EXTFLASH\_E~~ | | ~~外付けフラッシュメモリ消去のみを行う。~~ | |
|  | ~~EX\_M\_EXTFLASH\_WR~~ | | ~~外付けフラッシュメモリへの書込みを行う。~~ | |
|  | ~~EX\_M\_EXTFLASH\_END\_WR~~ | | ~~外付けフラッシュダウンロードの終了フラグ。上記2つのデファインとORして指定する。~~ | |
|  | | 注:本開発バージョンでは、外付けフラッシュをサポートしない | | |
|  | |  | | |
| addr\_1 | | | | | |
|  | | <address value> | | 書き込み開始アドレス | |
|  | |  | |  | |
| addr\_2 | | | | | |
|  | | <address value> | | ベリファイエラー発生時のアドレス | |
|  | |  | |  | |
| acc\_size | | | | | |
|  | | EX\_DATA\_SIZE\_1 | | 1バイトアクセス | |
|  | | EX\_DATA\_SIZE\_2 | | 2バイトアクセス | |
|  | | EX\_DATA\_SIZE\_4 | | 4バイトアクセス | |
|  | | EX\_DATA\_SIZE\_8 | | 8バイトアクセス | |
|  | |  | |  | |
| \*data\_1 | | | | | |
|  | | <0-FFh> | | 書き込むデータ | |
|  | |  | |  | |
| count | | | | | |
|  | | <1-FFFFh> | | 書き込みデータのアクセスサイズ単位の個数 | |
|  | |  | |  | |
| verify\_fg | | | | | |
|  | | EX\_VER\_OFF | | ベリファイしない | |
|  | | EX\_VER\_ON | | ベリファイする | |
|  | |  | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| direct | | | |
|  | メモリアクセスのアクセス手段情報を格納する。 | | |
|  | 【RH850 G3世代MCUの場合】 | | |
|  | bit0-2 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 000b:  CPU経由でアクセスしていない。  001b～111b:  CPU経由でアクセス時のアクセスしたPE番号。  001bの場合、CPU@PE1のアクセスを示す。 |
|  | bit3-5 | EXECのアクセス手段2  MAU経由でアクセスしたときに使用したMAUのPE番号 | 000b:  MAU経由でアクセスしていない。  001b～111b:  MAU経由でアクセス時のアクセスしたPE番号。  011bの場合、MAU@PE3でのアクセスを示す。 |
|  | bit6 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit7 | 未定義 |  |
|  |  |  |  |
|  | 【RH850 G4世代MCUの場合】 | | |
|  | bit0-3 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 0000b:  CPU経由でアクセスしていない。  0001b～1111b:  CPU経由でアクセス時のアクセスしたPE番号。  0001bの場合、CPU@PE0のアクセスを示す。  ：  1000bの場合、CPU@PE7のアクセスを示す。 |
|  | bit4-7 | EXECのアクセス手段2  MAU経由でアクセスしたときのカレントPE番号 | 0000b:  MAU経由でアクセスしていない。  0001b～1111b:  MAU経由でアクセス時のカレントPE番号。  0001bの場合、PE0選択時にMAUアクセスの発生を示す。  ：  1000bの場合、PE7選択時にMAUアクセスの発生を示す。 |
|  | bit8 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit9-31 | 未定義 |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_UE\_VERIFY ベリファイエラー(addr\_2にエラーアドレスを返す)

EX\_FE\_M\_MONTIMOUT\_MEM メモリ関数モニタタイムアウト

EX\_UE\_FLMDLEVEL FLMD端子書き込み禁止状態

EX\_UE\_FLASHENV フラッシュ書き込み不可状態

EX\_UE\_BLKCHKFAILED ブランクチェック失敗

EX\_UE\_ERASEFAILED イレース失敗

EX\_UE\_WRITEFAILED ライト失敗

EX\_PE\_OVERBLOCKAREA フラッシュメモリのブロックを跨いだ指定をした

EX\_UE\_PLLUNLOCK フラッシュ書き込みのためのPLL駆動でロック状態にできなかった

EX\_DE\_M\_DCUACCESS DCUアクセス異常

EX\_UE\_DFLNOSPRT データフラッシュサポート外デバイス

EX\_UE\_DFLENV データフラッシュ以外のフラッシュ環境中

EX\_PE\_DFLRANGE データフラッシュの範囲を超えた指定をした

EX\_NOSUPRT 本関数は未サポート

EX\_SE\_UNLOCKFAILED セキュリティ解除に失敗した

EX\_DE\_MAU\_NOSUPRTAREA 指定したエリアは、MAUアクセスをサポートしていないためアクセスできない

EX\_PE\_OVERLAP\_MCURSC 指定したエリアは、MCUの資源を跨ぐためアクセスできない

EX\_SE\_ACC\_RUN\_PEMODE フラッシュ書き換えモード中(P/Eモード中)のため、フラッシュ資源の読み出しおよび、書き込みはできない

EX\_SE\_ACC\_FCUFSEL\_FCU FCUファーム領域選択がFCUファームウェア資源を指定しているため、読み出しおよび、書き込みはできない

EX\_SE\_ANOTHERPERUN 別のPEが実行中のためFlash書き込みできない

EX\_SE\_AXICLKDSB\_ACCFAILED AXIクロックが供給されてないため、XC領域にアクセスできない

EX\_SE\_SFLCLKDSB\_ACCFAILED Serial Flashにクロックが供給されていないため、Serial Flash領域にアクセスできない

EX\_SE\_SDRAMCLKDSB\_ACCFAILED SDRAMにクロックが供給されていないため、SDRAM領域にアクセスできない

EX\_SE\_NOSPRT\_RAMINITOFF RAM初期化がOFFのためFlash書き込みできない

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_DE\_SECAREAACCERR　　　　 指定された領域にセキュア領域が含まれるため選択されたコアではアクセスできない

EX\_SE\_NOSPRT\_DURING\_CANBOOT CANブートデバッグ中のため本関数は使用できない

EX\_DE\_CANTACC\_CFLSETTING　　アクセスできないコードフラッシュ配置に設定されている

EX\_SE\_CPUSTOP\_AREA\_ACCFAILED CPUが停止中であるため指定した領域にアクセスできない

EX\_SE\_CPUSTOP\_FLASH\_WRITEFAILED CPUが停止中であるためフラッシュ領域へライトできない

EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED 　Cyclicモード中であるためフラッシュ領域へライトできない

EX\_SE\_STANDBY\_NOSPRT 　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

書き込み開始アドレス(addr\_1)から、書き込みデータ(\*data)をアクセスサイズ(acc\_size)に従って、個数(count)分書き込む。

[引数説明]

modeは、指定のカテゴリごとに説明が分類される。

＜カテゴリ別の説明＞

1.ユーザプログラム実行中の処理

ブレーク中の場合は、実行状態による処理の相違なく書き込みを行う。

フラッシュメモリ、外付けフラッシュメモリ、もしくはユーザプログラム実行中にアクセスできない領域を指定した場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

2.フラッシュメモリ書き換え時の処理

ユーザプログラム実行中の場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

コードフラッシュ、および、データフラッシュ、フラッシュエキストラ領域以外の領域を指定した場合は、エラー(EX\_PE\_DATA)を返す。

フラッシュエキストラ領域がArea0とArea1に分かれており、どちらかがValidでどちらかがInvalidである仕様のデバイスに対して、ValidのAreaを指定した場合、エラー(EX\_PE\_DATA)を返す。

フラッシュエキストラ領域のTAG Areaを指定した場合、エラー(EX\_PE\_DATA)を返す。

また、データフラッシュ領域、フラッシュエキストラ領域を指定し、acc\_sizeがEX\_DATA\_SIZE\_8の場合は、エラー(EX\_PE\_DATA)を返し処理は行わない。

~~3.外付けフラッシュメモリ書き換え時の処理~~

~~ユーザプログラム実行中の場合は、エラー(EX\_SE\_USRPGMRUN)を返す。~~

~~外付けフラッシュメモリ以外の領域を指定した場合は、エラー(EX\_PE\_M\_NOEXTFLAREA)を返す。~~

注:本開発バージョンでは、外付けフラッシュをサポートしない

addr\_2は[ReturnValues]に記載されているエラーの内、エラーアドレスを返すとなっていないエラーの場合、何も返却されない。

verify\_fgにEX\_VER\_ONを設定した場合はベリファイを行う。

ベリファイエラーが発生した場合は、処理を中断しエラー(EX\_UE\_VERIFY)を返す。

ただし、mode=EX\_M\_DMM\_WRの場合、ベリファイチェックを無視する。

~~また、EX\_M\_FLASH\_E\_WRでベリファイ指定をした場合は、消去後ブランクチェックを行う。この場合、エラーアドレスが特定できないためaddr\_2が示す値は不定になる。~~

[デバッガ関係者向け注意事項]

書き込みデータのバイト数は、[acc\_sizeが示すバイト数 \* count]分のサイズとなる。このため、書き込みデータの最大サイズは以下の様になる。

acc\_size=EX\_DATA\_SIZE\_1 --> 1[byte] \* 65535(FFFFH)[個] = 65535[byte]( 64K - 1[byte])

acc\_size=EX\_DATA\_SIZE\_2 --> 2[byte] \* 65535(FFFFH)[個] = 131070[byte](128K - 2[byte])

acc\_size=EX\_DATA\_SIZE\_4 --> 4[byte] \* 65535(FFFFH)[個] = 262140[byte](256K - 4[byte])

acc\_size=EX\_DATA\_SIZE\_8 --> 8[byte] \* 65535(FFFFH)[個] = 524280[byte](512K - 8[byte])

SFR(周辺I/O)アクセス時に使用する場合、リドロウ対策の代替SFRを考慮したアドレスを指定する必要がある。

ECC無効化処理を行うため、デバイス仕様のメモリマッピングで定義されている領域ごとに本関数を呼び出す必要がある。複数の領域を跨ぐ指定を行わないこと。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

ホットプラグインRUN中(接続～強制ブレークするまで)は、mode=EX\_BREAK\_WRをEX\_SE\_USRPGMRUNエラー

とする。EX\_M\_DMM\_WRを利用すること。

[非同期デバッグモード時の注意点]

コードフラッシュ、データフラッシュ、フラッシュエキストラ領域への書き込みは全コアがブレークしている場合のみ使用可能。全コアがブレークしていない場合はエラー(EX\_SE\_ANOTHERPERUN)が返る。

[動作説明]

mode別の動作を説明する。

|  |  |
| --- | --- |
| <EX\_NONBR\_WR> | |
|  | ブレーク中のみ書き込み処理を行い、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)を返す。 |
|  | |
| <EX\_BREAK\_WR> | |
|  | EXEC関数をex\_stop() → ex\_memwrite() → ex\_run()の順に呼び出したのと同じ操作をする。  ブレークの後処理を完了した状態でデータを書き込む。 |
|  | |
| <EX\_M\_DMM\_WR> | |
|  | G3世代MCUの場合はLocal RAM領域、Global RAM領域、Retention RAM領域、Primary RAM領域、Secondary RAM領域のみ、G4世代MCUの場合はG3世代MCUの領域に加えてCluster RAM領域、周辺I/O領域をMAUを使用してアクセス可能。MAUはDMAと同様にユーザプログラム動作のWAITを発生させるため、完全にユーザプログラム実行が連続して行われる訳ではない。  上記以外の領域に対して、指定した場合はEX\_DE\_MAU\_NOSUPRTAREAを返す。  [デバッガ関係者向け注意事項]  D1xのRetention RAM領域は対象外(クロスコネクト領域内にあるため)。  周辺I/O領域はユーザによるアクセス制限が設定可能なため本開発バージョンではサポートしない。 |
|  | |
| <EX\_M\_FLASH\_PV\_WR、EX\_M\_FLASH\_E\_WR、EX\_M\_FLASH\_EP\_WR、EX\_M\_FLASH\_P\_WR> | |
|  | フラッシュメモリへの書き込みに使用する。フラッシュのプロセスによってはサポートできていない場合もあるので注意のこと。  フラッシュの書き込みは、ブロックサイズ分のデータを上位ソフト側で用意して本関数を呼び出すようにすること。ブロックサイズの情報は、ex\_socunitinfo2()で得ることができる。  フラッシュエキストラ領域は、Block Protection Area、Configuration Setting Area、Security Setting Area、Switch Area、TAG Areaの5つのエリアを指す。このエリアの書き換えはG4世代MCUのみサポートする。  消去指定(EX\_M\_FLASH\_E\_WR)の場合は、アドレスで対象ブロックを判断する。この時のデータ指定は、アクセスサイズは任意でよいが、count値は1以上にすること。また、書き込むデータには少なくともアクセスサイズに相当するバイト数の0xFFを設定すること。  デバイスファイルの情報でフラッシュメモリ非搭載、もしくはLocalRAMがLocalRAM実行処理で使用するサイズ以下の場合は、エラー(EX\_UE\_FLASHENV)を返す。  LocalRAM実行処理中に処理が終了しなかった場合、エラー(EX\_DE\_M\_DCUACCESS)を返す。このエラーが返る場合はデバッグモニタが正常に終了できていないため、ex\_resetによって復帰すること。  フラッシュメモリへの書き込みの際にはセキュリティの解除処理も伴う。このとき、セキュリティの解除に失敗した場合はエラー(EX\_SE\_UNLOCKFAILED)を返す。ex\_socunitinfo2のEX\_UNIT\_RSUでセキュリティIDコードが解除されているはずなので、ほとんど返ることはないと思われるが、ユーザがセルフプログラミングでセキュリティIDコードを変更した場合に返る可能性がある。この場合は、変更後のセキュリティIDコードで接続しなおす必要がある。  フラッシュメモリ書き換え処理は、EX\_M\_FLASH\_END\_WRがORされない場合は、リソース(汎用レジスタやLocalRAM)を使用した状態のままでex\_memwriteを完了させる。このため、必ずEX\_M\_FLASH\_END\_WRをORしてフラッシュ書き換え処理を終了させること。  また、何らかのエラーが起きた場合は、EX\_M\_FLASH\_END\_WRビットに関係なくリソースを開放する。このとき、高速化したクロックを元に戻すためリセットが発生する。EX\_M\_FLASH\_END\_WRをORしたモードで呼び出す前に他の操作をしないよう注意すること。 |

|  |  |
| --- | --- |
|  | modeの設定でコードフラッシュメモリ、データフラッシュメモリ、フラッシュエキストラ領域、外付けフラッシュメモリ、その他のメモリと5種類のエリアに対する書き込みが可能であるが、モードを変える場合はEX\_M\_FLASH\_END\_WRやEX\_M\_EXTFLASH\_END\_WR付きで処理した後に、別なモードに変更すること。  フラッシュエキストラ領域は、Block Protection Area、Configuration Setting Area、Security Setting Area、Switch Area、TAG Areaの5つのエリアに分かれるが、書き換えエリアを変更する場合は、EX\_M\_FLASH\_END\_WRで各エリアの書き換え処理を完了後に、別のエリアへ変更すること。  verify\_fgは、フラッシュ書き込み後に再度通常のリードを行い比較するかどうかの指定である。  ex\_socunitinfo2のEX\_UNIT\_LPDOPTで「RAM初期化をしない」に設定している場合、エラー(EX\_SE\_NOSPRT\_RAMINITOFF)を返し、処理は行わない。フラッシュメモリ書き換え内部処理でRAM領域への書き込みにより、意図せぬ初期化や、ECCエラーが発生してしまうため。フラッシュメモリの書き換えを行いたい場合は「RAM初期化をする」に設定すること。  フラッシュの書き込みにはFACI制御プログラムLRAMMONを使用する。LRAMMONはワークRAM領域に配置される。ワークRAM領域の情報は、ex\_socunitinfo2()で設定/参照することができる。  EX\_M\_FLASH\_PV\_WR指定でコードフラッシュ領域を書き換える場合は、書き換え対象のブロックを読み出した後、書き込みデータとマージし、フラッシュメモリブロック単位でのフラッシュ書き換えを行う。これにより書き込みデータのない領域は読み出したデータが書かれる。ブランクは0xFFが読めるため、0xFFが書かれる。  EX\_M\_FLASH\_EP\_WR指定でコードフラッシュ領域を書き換える場合は、フラッシュメモリブロック単位でのフラッシュ書き換えを行い、書き込みブロック内で書き込みデータのない領域に対しては0xFFでフィルする。  EX\_M\_FLASH\_P\_WR指定でコードフラッシュ領域を書き換える場合は、FCU書き込み単位でのフラッシュ書き換えを行う。書き込みブロック内で書き込みデータのない領域はブランクのままにする。書き込みデータのサイズがFCU書き込み単位の倍数でなかった場合、FCU書き込み単位に自動的に補正し、端数領域は0xFFでフィルする。  各フラッシュエキストラ領域がArea0とArea1に別れており、どちらかがValidでどちらかがInvalidである仕様のデバイスでは、InvalidのAreaを指定し書き換える場合、自動でSwitch AreaとTAG Areaを操作し、ValidのAreaとInvalidのAreaを切り替える。そのため、書き換え時にInvalidのAreaに書き込んだデータは、書き換え後にはValidのAreaに反映される。同時に、書き換え前のValidのAreaは、書き換え後にはInvalidのAreaになる。  ValidのAreaとInvalidのAreaを切り替えるのは書き換え指定されたフラッシュエキストラ領域のみであり、他のフラッシュエキストラ領域は書き換え前のままとなる。  Switch AreaのInvalid Areaを指定し書き換える場合、自動でTAG Areaを操作し、書き換えたSwitch Areaの設定に合わせて各フラッシュエキストラ領域のValidのAreaとInvalidのAreaが決定される。  [デバッガ関係者向け注意事項]  (1)EX\_M\_FLASH\_P\_WRを指定したFCU書き込み単位でのフラッシュ書き換え後、ソフトブレーク機能やメモリパネルからのフラッシュメモリ操作によってEX\_M\_FLASH\_PV\_WRまたはEX\_M\_FLASH\_EP\_WRを指定したフラッシュ書き換えを行なった場合、書き換えたフラッシュメモリブロックのブランクは0xFFでフィルされる。  (2)EX\_M\_FLASH\_P\_WRを指定したFCU書き込み単位でのフラッシュ書き換えを行う場合、一度のフラッシュ書き換えで同じFCU書き込み単位領域に2回以上書き込みを行なわないこと。 |
|  |  |

|  |  |
| --- | --- |
|  | modeに以下のデファインを指定した場合、CPUクロックを一時的に変更して処理し、処理後元に戻す。(将来、CPUクロックの復帰が難しいCPUが現れた場合は、CPUリセットをする可能性がある)  ＜対象となるmode＞  EX\_M\_FLASH\_PV\_WR  EX\_M\_FLASH\_E\_WR  EX\_M\_FLASH\_EP\_WR  EX\_M\_FLASH\_P\_WR  (EX\_M\_FLASH\_END\_WRをORする場合も含む)  [デバッガ関係者向け注意事項]  クロック設定機能に依存し、CPUリセットが必要かどうかは動作確認後に決定する |
|  |  |
|  | ＜フラッシュメモリへの書き込みに使用するモードの動作＞  ・EX\_M\_FLASH\_PV\_WR  処理内容： 元のデータをモディファイして対象データを書き込む  処理順序： Read - Erase - Modify - Program - (rVerify) 対象:コード/データフラッシュ  Read - Modify - Config Setting - (rVerify) 対象:フラッシュエキストラ領域  注：書き込み対象によって、処理順序は異なる。  ・EX\_M\_FLASH\_E\_WR  処理内容： フラッシュメモリを消去状態にする  処理順序： Erase  ・EX\_M\_FLASH\_EP\_WR  処理内容： 元のデータを無視して対象データを書き込む  処理順序： Erase - Program - (rVerify)  ・EX\_M\_FLASH\_P\_WR  処理内容： フラッシュメモリに対象データを書き込む  処理順序： Program - (rVerify)  処理の説明  Read フラッシュメモリの内容を読み出して、バッファへ格納する  Erase 消去する  Modify 書き込みデータをバッファへ書き込む  Program バッファの内容をフラッシュメモリへ書き込む  　Config Setting バッファの内容をフラッシュエキストラ領域へ書き込む  (rVerify) リードベリファイ(ベリファイフラグがある場合実行)  ＜データフラッシュ搭載デバイスのmode指定＞  EX\_M\_FLASH\_PV\_WR、EX\_M\_FLASH\_E\_WRは、コードフラッシュおよびデータフラッシュに対する書き込みに使用する。コードフラッシュおよびデータフラッシュのアドレス範囲を自動判定するが、異なる種類の領域を同時に指定することはできない。  [デバッガ関係者向け注意事項]  EX\_M\_FLASH\_P\_WRを指定したFCU書き込み単位でのフラッシュ書き換え前に、EX\_M\_FLASH\_E\_WR指定で対象領域を消去すること。 |

|  |  |
| --- | --- |
|  | ・mode指定概要  1.EX\_M\_FLASH\_PV\_WR  1-1.addr\_1に指定したアドレスがデータフラッシュ領域  --> データを書き込む  1-2.addr\_1に指定したアドレスがコードフラッシュ領域  --> データを書き込む  1-3.addr\_1に指定したアドレスがフラッシュエキストラ領域  --> データを書き込む  1-4.addr\_1に指定したアドレスが上記のいずれでもない  -->エラー(EX\_PE\_DATA)となる  2.EX\_M\_FLASH\_E\_WR  2-1.addr\_1に指定したアドレスがデータフラッシュ領域  --> データを消去する  2-2.addr\_1に指定したアドレスがコードフラッシュ領域  --> データを消去する  2-3.addr\_1に指定したアドレスが上記のいずれでもない  --> エラー(EX\_PE\_DATA)となる  3.EX\_M\_FLASH\_EP\_WR  3-1.addr\_1に指定したアドレスがデータフラッシュ領域  --> エラー(EX\_PE\_DATA)となる  3-2.addr\_1に指定したアドレスがコードフラッシュ領域  --> データを書き込む  3-3.addr\_1に指定したアドレスが上記のいずれでもない  --> エラー(EX\_PE\_DATA)となる  4.EX\_M\_FLASH\_P\_WR  4-1.addr\_1に指定したアドレスがデータフラッシュ領域  --> エラー(EX\_PE\_DATA)となる  4-2.addr\_1に指定したアドレスがコードフラッシュ領域  --> データを書き込む  4-3.addr\_1に指定したアドレスが上記のいずれでもない  --> エラー(EX\_PE\_DATA)となる  5.EX\_M\_FLASH\_END\_WRをORする場合  フラッシュ書き込みはmodeにEX\_M\_FLASH\_END\_WRをORして完了させる必要がある。  このため、EX\_M\_FLASH\_END\_WRをORした指定を行わずにex\_memwriteの呼び出しの繰り返しを中断して他の関数を呼び出さないこと。  Phase2.1まではEX\_M\_FLASH\_END\_WRをORしたmode指定を行うまで、一括書き込みを行わなかったが、Phase2.2以降は4byte単位書き込み方式になるため、ex\_memwriteの呼び出しごとに書き込みを行う。 |

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

CANブートデバッグ時に本コマンドでFlash資源にアクセスすると、エラー(EX\_SE\_NOSPRT\_DURING\_CANBOOT)が返る。

G4世代デバイスでは、セキュリティコアを有効にした動作モード時、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置に切り替わっている可能性がある。

そのため、セキュリティコアを有効にした動作モードでかつ、ユーザのデバイス設定によってコードフラッシュ領域にアクセスできない配置に変更されていた場合、エラー(EX\_DE\_CANTACC\_CFLSETTING)を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・操作対象のPEが停止(初期停止・Cyclic Disable)中で、アクセス領域のメモリアクセス手段がCPUアクセスの場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_AREA\_ACCFAILED)する。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

・デバイスがCyclicモード中は、デバイス仕様によりフラッシュ領域の書き換えを行うことができない。Cyclicモード中にフラッシュの書き換えを指定した場合は、書き込みを行わずエラー応答(EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED)する。

・メインコアデバッグ機能と初期停止・スタンバイモードデバッグ環境を同時に有効にし、G3世代デバイスではPE1(G4世代デバイスではPE0)が初期停止状態である場合、フラッシュ領域の書き換えを行うことができない。フラッシュの書き換えを指定した場合は、書き込みを行わずエラー応答(EX\_SE\_CPUSTOP\_FLASH\_WRITEFAILED)する。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[FCUに関連する注意事項]

・メモリアクセスするフラッシュ資源がP/Eモード中は、メモリアクセスを行わずEX\_SE\_ACC\_RUN\_PEMODEを返送する。

・FCUファーム領域選択レジスタがFCUファームウェア資源を指定し、コードフラッシュを対象としてメモリアクセスを行った場合は、メモリアクセスを行わずEX\_SE\_ACC\_FCUFSEL\_FCUを返送する。

[フラッシュメモリのBank跨ぎの注意点]

modeにEX\_M\_FLASH\_END\_WRをORせずにコードフラッシュを連続モードで書き込む場合、DualBankのBankを跨いでの消去・書き込みは可能。データフラッシュに関しては、modeにEX\_M\_FLASH\_END\_WRをORせずに連続モードでDualBankのBankを跨いで書き込みを行った場合、エラー(EX\_PE\_DATA)が返る。データフラッシュはDualBankのBankを跨ぐ場合は、Bankを跨がないブロックの位置で一旦EX\_M\_FLASH\_END\_WRをmodeにORして消去・書き込みを終了させてから、次のBankのブロック位置から消去・書き込みを行うこと。データフラッシュは高速化のため、EX\_M\_FLASH\_END\_WR付きmodeで呼ばれたときに一括で消去・書き込みを行う機能があり、途中でFPSYSの切り替えができないため。

[EraseCounterの注意点]

DataFlashの先頭からの11ブロックがEraseCounterに割り当てられている品種がある。EraseCounterの領域に対しては、書き込み・消去ができないため注意すること。

＜フラッシュメモリの書き込み単位＞

(1)コードフラッシュ

・EX\_M\_FLASH\_E\_WR

最小消去単位がブロック単位のため、1ブロック単位で消去する。

・EX\_M\_FLASH\_PV\_WR

消去をブロック単位で行い、書き込みデータのサイズが1ブロック未満の場合でも1ブロック単位で書き込む。書き込み対象ブロック内で書き込みデータのない領域は消去前の値を再度書き込む(ブランク状態だった領域は0xFFが読めるため、0xFFを書き込む)。

・EX\_M\_FLASH\_EP\_WR

消去をブロック単位で行い、書き込みデータのサイズが1ブロック未満の場合でも1ブロック単位で書き込む。書き込み対象ブロック内で書き込みデータのない領域は0xFFでフィルする。

・EX\_M\_FLASH\_P\_WR

消去は行なわず、FCUによるフラッシュプログラミングの書き込み単位で書き込む。書き込み対象ブロック内で書き込みデータのない領域はブランクのままとする(RV40Fの場合、コードフラッシュのブランク領域を読み出そうとした場合、ECCエラーになる)。

書き込みアドレスがFCU書き込み単位の境界にない場合は、FCU書き込み単位に満たないデータを0xFFでフィルしてFCU書き込み単位境界のデータを作成して書き込む。

例) 書き込みパターンと書き込み後のメモリダンプイメージの例

(指定modeはEX\_M\_FLASH\_P\_WR、FCU書き込み単位は256byteとする)

① ブランク状態の0x00010000番地へ256byteデータ(0x5A5A5A5A…)を書き込み

② ブランク状態の0x00010000番地へ254byteデータ(0x5A5A5A5A…)を書き込み

③ ブランク状態の0x00010000番地へ1byteデータ(0x5A)を書き込み

④ ブランク状態の0x000100FF番地へ2byteデータ(0x5A5A)を書き込み

|  |  |  |  |
| --- | --- | --- | --- |
| No | アドレス | 書き込み前データ | 書き込み後データ |
| ① | 0x00010000 | 00 01 … FD FE FF 00 01 02 …  :\*\* \*\* … \*\* \*\* \*\* \*\* \*\* \*\* … | 00 01 … FD FE FF 00 01 02 …  :5A 5A … 5A 5A 5A \*\* \*\* \*\* … |
| ② | 0x00010000 | 00 01 … FD FE FF 00 01 02 …  :\*\* \*\* … \*\* \*\* \*\* \*\* \*\* \*\* … | 00 01 … FD FE FF 00 01 02 …  :5A 5A … 5A FF FF \*\* \*\* \*\* … |
| ③ | 0x00010000 | 00 01 … FD FE FF 00 01 02 …  :\*\* \*\* … \*\* \*\* \*\* \*\* \*\* \*\* … | 00 01 … FD FE FF 00 01 02 …  :5A FF … FF FF FF \*\* \*\* \*\* … |
| ④ | 0x00010000 | 00 01 … FD FE FF 00 01 02 …  :\*\* \*\* … \*\* \*\* \*\* \*\* \*\* \*\* … | 00 01 … FD FE FF 00 01 02 …  :FF FF … FF FF 5A 5A FF FF … |

注：\*はブランク状態を示す(ブランク領域を読み出すとECCエラー)

(2)データフラッシュ

・EX\_M\_FLASH\_E\_WR

最小消去単位がブロック単位のため、1ブロック単位で消去する。

・EX\_M\_FLASH\_PV\_WR

消去をブロック単位で行い、書き込みは最小書き込み単位の4byte単位で行う。

書き込みアドレスが4byte境界にない場合は、4byteに満たないデータを0xFFでフィルして4byte境界のデータを作成して書き込む。また、書き込みデータが1ブロックサイズに満たない場合、元々書き込まれていたデータがある場合はそのデータを復元する。ブランク状態であった領域はブランク状態を保持する(RV40Fの場合、データフラッシュのブランク状態は不定値)。

注：Phase2.1までは1ブロック単位で(書き込み指定データのない領域は0xFFでフィルしたデータで)の書き込みを行っていたが、Phase2.2以降は4byte単位での書き込みを行う。

例) 書き込みパターンと書き込み後のメモリダンプイメージの例

① 非ブランク状態の0xFF200000番地へ4byteデータ(0x5A5A5A5A)を書き込み

② 非ブランク状態の0xFF200000番地へ2byteデータ(0x5A5A)を書き込み

③ 非ブランク状態の0xFF200002番地へ1byteデータ(0x5A)を書き込み

④ ブランク状態の0xFF200000番地へ4byteデータ(0x5A5A5A5A)を書き込み

⑤ ブランク状態の0xFF200003番地へ2byteデータ(0x5A5A)を書き込み

⑥ ブランク状態の0xFF200002番地へ1byteデータ(0x5A)を書き込み

⑦ ブランク状態と非ブランク状態混在の0xFF200003番地へ2byteデータ(0x5A5A)を書き込み

|  |  |  |  |
| --- | --- | --- | --- |
| No | アドレス | 書き込み前データ | 書き込み後データ |
| ① | 0xFF200000 | :00 01 02 03 04 05 06 07 08 … | :5A 5A 5A 5A 04 05 06 07 08 … |
| ② | 0xFF200000 | :00 01 02 03 04 05 06 07 08 … | :5A 5A 02 03 04 05 06 07 08 … |
| ③ | 0xFF200000 | :00 01 02 03 04 05 06 07 08 … | :00 01 5A 03 04 05 06 07 08 … |
| ④ | 0xFF200000 | :\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* … | :5A 5A 5A 5A \*\* \*\* \*\* \*\* \*\* … |
| ⑤ | 0xFF200000 | :\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* … | :FF FF FF 5A 5A FF FF FF \*\* … |
| ⑥ | 0xFF200000 | :\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* … | :FF FF 5A FF \*\* \*\* \*\* \*\* \*\* … |
| ⑦ | 0xFF200000 | :00 01 02 03 \*\* \*\* \*\* \*\* \*\* … | :00 01 02 5A 5A FF FF FF \*\* … |

注：\*はブランク状態を示す(ブランク状態は不定値)

(3)フラッシュエキストラ領域

・EX\_M\_FLASH\_PV\_WR

フラッシュエキストラ領域はブランク状態が存在せず、必ずデータが書き込まれている状態からの書き換え処理となる。書き換えは特定の一部エリアを除き最小書き込み単位の4byte単位で行う。

書き込みアドレスが4byte境界にない場合や、書き込みデータが4byteに満たない場合は元々書き込まれていたデータから4byteの書き換えデータを作成し書き込みを行う。

例) 書き込みパターンと書き込み後のメモリダンプイメージの例

① 0xFF320080番地へ4byteデータ(0x5A5A5A5A)を書き込み

② 0xFF320080番地へ2byteデータ(0x5A5A)を書き込み

③ 0xFF320082番地へ1byteデータ(0x5A)を書き込み

④ 0xFF320083番地へ2byteデータ(0x5A5A)を書き込み

|  |  |  |  |
| --- | --- | --- | --- |
| No | アドレス | 書き込み前データ | 書き込み後データ |
|  | 0xFF320080 | :00 01 02 03 04 05 06 07 08 … | :5A 5A 5A 5A 04 05 06 07 08 … |
|  | 0xFF320080 | :00 01 02 03 04 05 06 07 08 … | :5A 5A 02 03 04 05 06 07 08 … |
|  | 0xFF320080 | :00 01 02 03 04 05 06 07 08 … | :00 01 5A 03 04 05 06 07 08 … |
|  | 0xFF320080 | :00 01 02 03 04 05 06 07 08 … | :00 01 02 5A 5A 05 06 07 08 … |

フラッシュエキストラ領域は基本的に4byte単位で書き換えを実施するが、Security Settings Area内のID格納エリアは256bitで構成されており、仮に4byte単位で書き換え中、256bit分の書き換えが終わる前に瞬間的な供給電源の遮断等により、書き換えが中断されてしまった場合、再認証が困難となる可能性がある。そこで、ID格納エリアだけは32byte単位で書き換えを実施し、書き換え中断時も256bit全てが書き換え前の状態へ戻るようにする。このIDエリアは、MCUのPRDSEL情報から領域判定し、32byte境界でリードモディファイライト処理を行う。

U2Aではフラッシュエキストラ領域はすべて32byte単位書き換えで行うデバイス仕様となった。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memfill | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

指定したアドレス範囲のメモリを初期化する

[Format]

int ex\_memfill( EXMEM \*fil )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*fil | <IO> | メモリアクセス情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <I > | 操作モード | ○ |
|  | ULONG | addr\_1 | <I > | 開始アドレス | ○ |
|  | ULONG | addr\_2 | < O> | ベリファイエラーアドレス | ○ |
|  | ULONG | number | <I > | 範囲のバイト数 | ○ |
|  | USHORT | acc\_size | <I > | アクセスサイズ | ○ |
|  | UCHAR | \*data\_1 | <I > | データバッファポインタ | ○ |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <I > | 初期化データのアクセスサイズ単位の個数 | ○ |
|  | USHORT | verify\_fg | <I > | ベリファイフラグ | ○ |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | < O> | メモリアクセス手段 | ○ |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | EX\_NONBR\_WR | ユーザプログラム実行中に呼び出された時、エラー(EX\_SE\_USRPGMRUN)とする |
|  | EX\_BREAK\_WR | ユーザプログラム実行中に呼び出された時、ex\_stop処理してからデータを書き込む。その後、プログラムを再実行する |
|  |  |  |
| addr\_1 | | |
|  | <address value> |  |
|  |  |  |
| addr\_2 | | |
|  | <address value> |  |
|  |  |  |
| number | | |
|  | <00000001 - ffffffffh> |  |
|  |  |  |
| acc\_size | | |
|  | EX\_DATA\_SIZE\_1 | 1バイトアクセス |
|  | EX\_DATA\_SIZE\_2 | 2バイトアクセス |
|  | EX\_DATA\_SIZE\_4 | 4バイトアクセス |
|  | EX\_DATA\_SIZE\_8 | 8バイトアクセス |
|  |  |  |
| \*data\_1 | | |
|  | <00000000 - ffffffffh> |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| count | | |
|  | <0001 - 0040h> | 設定できる初期化データは最大64個とする |
|  |  |  |
| verify\_fg | | |
|  | EX\_VER\_OFF | ベリファイしない |
|  | EX\_VER\_ON | ベリファイする |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Direct | | | |
|  | メモリアクセスのアクセス手段情報を格納する。 | | |
|  | 【RH850 G3世代MCUの場合】 | | |
|  | bit0-2 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 000b:  CPU経由でアクセスしていない。  001b～111b:  CPU経由でアクセス時のアクセスしたPE番号。  001bの場合、CPU@PE1のアクセスを示す。 |
|  | bit3-5 | EXECのアクセス手段2  MAU経由でアクセスしたときに使用したMAUのPE番号 | 000b:  MAU経由でアクセスしていない。  001b～111b:  MAU経由でアクセス時のアクセスしたPE番号。  011bの場合、MAU@PE3でのアクセスを示す。 |
|  | bit6 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit7 | 未定義 |  |
|  |  |  |  |
|  | 【RH850 G4世代MCUの場合】 | | |
|  | bit0-3 | EXECのアクセス手段1  CPU経由でメモリアクセスしたときに使用したCPUのPE番号 | 0000b:  CPU経由でアクセスしていない。  0001b～1111b:  CPU経由でアクセス時のアクセスしたPE番号。  0001bの場合、CPU@PE0のアクセスを示す。  :  1000bの場合、CPU@PE7のアクセスを示す。 |
|  | bit4-7 | EXECのアクセス手段2  MAU経由でアクセスしたときのカレントPE番号 | 0000b:  MAU経由でアクセスしていない。  0001b～1111b:  MAU経由でアクセス時のカレントPE番号。  0001bの場合、PE0選択時にMAUアクセスの発生を示す。  ：  1000bの場合、PE7選択時にMAUアクセスの発生を示す。 |
|  | bit8 | 指定コア空間内の内蔵RAM資源を  MAUアクセスする場合、EXEC内部で  アドレス変換処理を実行したかを  格納する情報 | 0: EXECでアドレス変換を行っていない。  1: EXECでアドレス変換を行った。 |
|  | bit9-31 | 未定義 |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_UE\_VERIFY ベリファイエラー

EX\_PE\_INIAREA 初期化データ数が初期化領域を超えている

EX\_FE\_M\_MONTIMOUT\_MEM メモリ関数モニタタイムアウト

~~EX\_UE\_BROMAREA 裏ROMエリアを指定した~~

EX\_PE\_FLASHACC\_NOSUPRT フラッシュメモリへのアクセスはサポートしていない

EX\_PE\_OVERLAP\_MCURSC 指定したエリアは、MCUの資源を跨ぐためアクセスできない

EX\_SE\_AXICLKDSB\_ACCFAILED AXIクロックが供給されてないため、XC領域にアクセスできない

EX\_SE\_SFLCLKDSB\_ACCFAILED Serial Flashにクロックが供給されていないため、Serial Flash領域にアクセスできない

EX\_SE\_SDRAMCLKDSB\_ACCFAILED SDRAMにクロックが供給されていないため、SDRAM領域にアクセスできない

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_DE\_SECAREAACCERR　　　　 指定された領域にセキュア領域が含まれるため選択されたコアではアクセスできない

EX\_SE\_CPUSTOP\_AREA\_ACCFAILED CPUが停止中であるため指定した領域にアクセスできない

EX\_SE\_STANDBY\_NOSPRT 　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

addr\_1のメモリ空間からnumberアドレス分のメモリを、data\_1が指し示すcount個のデータで繰り返し初期化する。

[引数説明]

numberには初期化対象範囲のバイト数を設定する。

\*data\_1に設定できる初期化データは最大512バイトまでとなる。

countに設定可能な初期化データの最大個数は64個である。

verify\_fgにEX\_VER\_ONが指定された場合は、メモリライト時にベリファイ動作を行い、ベリファイエラーが発生した場合、処理を中断しaddr\_2にベリファイエラーアドレスを返す。

[動作説明]

numberが、アクセスサイズで割り切れない場合、余ったデータを無視する。

addr\_1およびnumberで設定した初期化領域(ただし初期化領域が64バイト以下の場合)を設定した初期化

データの最大数が上回った場合、エラー(EX\_PE\_INIAREA)を返し処理を行わない。

コードフラッシュ、データフラッシュに対しての書き込みが発生した場合は、エラー(EX\_PE\_FLASHACC\_NOSUPRT)を返す。

ホットプラグインRUN中(接続～強制ブレークするまで)は、mode=EX\_BREAK\_WRをEX\_SE\_USRPGMRUNエラー

とする。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

G4世代のFlash領域へのメモリアクセスでは、選択されているコアから指定された領域にアクセスする。

指定された領域にセキュア領域が含まれていてかつ、選択されているコアがセキュア領域にアクセスできない場合、エラー(EX\_DE\_SECAREAACCERR)を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・操作対象のPEが初期停止状態またはCyclic Disable状態で、アクセス領域のメモリアクセス手段がデバッグモニタの場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_AREA\_ACCFAILED)する。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memcopy | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **×** | **TRC1** | **／** | **TRC2** | **／** |

[Function]

指定したアドレス範囲のメモリの内容をコピーする

[Format]

int ex\_memcopy( EXMEM \*cpy )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*cpy | <IO> | メモリアクセス情報 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <--> |  | -- |
|  | ULONG | addr\_1 | <I > | 転送元アドレス | ○ |
|  | ULONG | addr\_2 | <IO> | 転送先アドレスもしくはベリファイエラーアドレス | ○ |
|  | ULONG | number | <I > | 範囲のバイト数 | ○ |
|  | USHORT | acc\_size | <I > | アクセスサイズ | ○ |
|  | UCHAR | \*data\_1 | <--> |  | -- |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <--> |  | -- |
|  | USHORT | verify\_fg | <I > | ベリファイフラグ | ○ |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | <--> |  | -- |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| addr\_1 | | |
|  | <address value> |  |
|  |  |  |
| addr\_2 | | |
|  | <address value> |  |
|  |  |  |
| number | | |
|  | <00000001 - 7fffffffh> |  |
|  |  |  |
| acc\_size | | |
|  | EX\_DATA\_SIZE\_1 | 1バイトアクセス |
|  | EX\_DATA\_SIZE\_2 | 2バイトアクセス |
|  | EX\_DATA\_SIZE\_4 | 4バイトアクセス |
|  | EX\_DATA\_SIZE\_8 | 8バイトアクセス |
|  |  |  |
| verify\_fg | | |
|  | EX\_VER\_OFF | ベリファイしない |
|  | EX\_VER\_ON | ベリファイする |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_UE\_VERIFY ベリファイエラー

EX\_FE\_M\_MONTIMOUT\_MEM メモリ関数モニタタイムアウト

~~EX\_UE\_BROMAREA 裏ROMエリアを指定した~~

EX\_PE\_OVERLAP\_MCURSC 指定したエリアは、MCUの資源を跨ぐためアクセスできない

EX\_SE\_AXICLKDSB\_ACCFAILED AXIクロックが供給されてないため、XC領域にアクセスできない

EX\_SE\_SFLCLKDSB\_ACCFAILED Serial Flashにクロックが供給されていないため、Serial Flash領域にアクセスできない

EX\_SE\_SDRAMCLKDSB\_ACCFAILED SDRAMにクロックが供給されていないため、SDRAM領域にアクセスできない

EX\_SE\_ANOTHERPERUN 別のPEが実行中のため使用できない

EX\_DE\_SECAREAACCERR　　　　 指定された領域にセキュア領域が含まれるため選択されたコアではアクセスできない

EX\_SE\_CPUSTOP\_AREA\_ACCFAILED CPUが停止中であるため指定した領域にアクセスできない

[Description]

addr\_1のメモリ空間からnumberアドレス分のメモリ内容を、addr\_2の領域に転送する。

[引数説明]

numberはaddr\_1もしくはaddr\_2からアドレス最大値を超えない値でなければならない。

(addr\_1 + number <= アドレス最大値 かつaddr\_2 + number <= アドレス最大値)

verify\_fgにEX\_VER\_ONが指定された場合は、メモリライト時にベリファイ動作を行い、ベリファイエラーが発生した場合、処理を中断しaddr\_2にベリファイエラーアドレスを返す。

[動作説明]

numberがアクセスサイズで割り切れない場合、余ったデータを無視する。

ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)を返し、処理は行わない。

コピー先の資源がコードフラッシュまたは、データフラッシュの場合は、エラー(EX\_PE\_FLASHACC\_NOSUPRT)を返す。

G4世代のFlash領域へのメモリアクセスでは、選択されているコアから指定された領域にアクセスする。

指定された領域にセキュア領域が含まれていてかつ、選択されているコアがセキュア領域にアクセスできない場合、エラー(EX\_DE\_SECAREAACCERR)を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

操作対象のPEが停止(初期停止・Cyclic Disable)中で、アクセス領域のメモリアクセス手段がCPUアクセスの場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_AREA\_ACCFAILED)する。

メモリアクセス手段の詳細は、「RH850G3/G4コア用E1/E20/E2/IE850A/IE850エミュレータF/W機能仕様書」を参照のこと。

[デバッガ関係者向け注意事項]

ECC無効化処理を行うため、デバイス仕様のメモリマッピングで定義されている領域ごとに本関数を呼び出す必要がある。複数の領域を跨ぐ指定を行わないこと。

[非同期デバッグモード時の注意点]

全コアがブレークしている場合のみ使用可能。全コアがブレークしていない場合はエラー(EX\_SE\_ANOTHERPERUN)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memsearch **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

指定したアドレス範囲のメモリを検索する

[Format]

int ex\_memsearch( EXMEM \*sch )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*sch | <--> | メモリアクセス情報 | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <--> |  | -- |
|  | ULONG | addr\_1 | <--> |  | -- |
|  | ULONG | addr\_2 | <--> |  | -- |
|  | ULONG | number | <--> |  | -- |
|  | USHORT | acc\_size | <--> |  | -- |
|  | UCHAR | \*data\_1 | <--> |  | -- |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <--> |  | -- |
|  | USHORT | verify\_fg | <--> |  | -- |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | <--> |  | -- |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| addr\_1 | | |
|  |  |  |
| addr\_2 | | |
|  |  |  |
| number | | |
|  |  |  |
| acc\_size | | |
|  |  |  |
| \*data\_1 | | |
|  |  |  |
| count |  |  |
|  |  |  |
| error\_fg | | |
|  |  |  |
| direct | | |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_memcompare **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

指定したアドレス範囲のメモリの内容を比較する

[Format]

int ex\_memcompare( EXMEM \*cmp )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXMEM | \*cmp | <--> | メモリアクセス情報 | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | UCHAR | mode | <--> |  | -- |
|  | ULONG | addr\_1 | <--> |  | -- |
|  | ULONG | addr\_2 | <--> |  | -- |
|  | ULONG | number | <--> |  | -- |
|  | USHORT | acc\_size | <--> |  | -- |
|  | UCHAR | \*data\_1 | <--> |  | -- |
|  | UCHAR | \*data\_2 | <--> |  | -- |
|  | USHORT | count | <--> |  | -- |
|  | USHORT | verify\_fg | <--> |  | -- |
|  | UCHAR | error\_fg | <--> |  | -- |
|  | USHORT | direct | <--> |  | -- |
| }EXMEM | | |  |  |  |

[Arguments]

|  |  |  |  |
| --- | --- | --- | --- |
| addr\_1 | | | |
|  |  |  | |
| addr\_2 | | | |
|  |  |  | |
| number | | | |
|  |  |  | |
| acc\_size | | | |
|  |  |  | |
| \*data\_1 | | | |
|  |  |  | |
| count | | | |
|  |  | |  |
| error\_fg | | | |
|  |  | |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

### レジスタ操作

レジスタ関数には、レジスタ・データの読み出しと書き込みを制御するレジスタ制御系関数がある。

下記に関数の一覧を示す。

レジスタ制御系関数

1)レジスタ・データの読み出し

ex\_regread

2)レジスタ・データの書き込み

ex\_regwrite

＜レジスタ制御系関数の注意事項＞

・システムレジスタのsystem register group 3は、ユーザ非公開であるため上位側ソフトで使用できない。また、使用してもエラーは発生しない。(EXEC内部処理で使用するため使用した場合の動作保障はない)

・汎用レジスタのR6、R7、R8、R9はEXEC内部処理で使用するため、ex\_regwriteで書き込みを行っても即時にCPUリソースに反映されない。その他の汎用レジスタは即時にCPUリソースに反映される。

・コプロセッサ関連レジスタは、搭載しないデバイスの場合、ブレーク中でも抑制不可能なUCPOP例外(コプロセッサ使用不可例外)の対象となってしまうためエラーを返し処理は行わない。

1)FPU関連レジスタは、搭載しないデバイスの場合、エラー(EX\_PE\_DATA)を返し処理は行わない。

|  |  |
| --- | --- |
| ＜FPU機能システムレジスタ＞ | |
| レジスタ名 | 機能 |
| FPSR | 浮動小数点演算の設定／ステータス |
| FPEPC | 浮動小数点演算例外プログラムカウンタ |
| FPST | 浮動小数点のステータス |
| FPCC | 浮動小数点演算の比較結果 |
| FPCFG | 浮動小数点機能の設定 |
| FPEC | 浮動小数点演算例外の制御 |

2)SIMD関連レジスタは、搭載デバイスがないため本開発バージョンではサポートしない。このため、エラー(EX\_PE\_DATA)を返し処理は行わない。

|  |  |
| --- | --- |
| ＜SIMD機能ベクトルレジスタ＞ | |
| レジスタ名 | 機能 |
| vr0-vr31 | SIMD演算ユニット専用のデータ・レジスタ |

|  |  |
| --- | --- |
| ＜SIMD機能システムレジスタ＞ | |
| レジスタ名 | 機能 |
| SESR | SIMD演算の設定／ステータス |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_regread | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

レジスタのデータを読み出す

[Format]

int ex\_regread( USHORT mode, USHORT num, USHORT \*red\_char, ULONG \*data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 操作モード | ○ |
| USHORT | num | <I > | 対象レジスタ個数 | ○ |
| USHORT | \*reg\_char | <I > | レジスタID(複数個可) | ○ |
| ULONG | \*data | < O> | アクセスデータ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | EX\_ER\_ON\_REG\_READ | ユーザプログラム実行中に呼び出された時、エラー(EX\_SE\_USRPGMRUN)とする。 |
|  | EX\_ER\_OFF\_REG\_READ | ユーザプログラム実行中に呼び出された時、ex\_stop処理をしてからレジスタを読み出す。読み込み後再実行する。 |
|  | ~~EX\_M\_DMM\_REG\_READ~~ | ~~ユーザプログラム実行中に呼び出された時、一瞬ブレークさせてデータを読み込む。~~ |
|  | 注：RH850ではEX\_M\_DMM\_REG\_READはサポートしない | |
| Num | | |
|  | <0001 - 0020h> | 読み出すレジスタ個数を指定する。 |
|  |  |  |
| \*reg\_char | | |
|  | <register ID> | レジスタ番号(複数個設定可) |
|  |  |  |
| \*data | | |
|  | <00000000 - ffffffffh> |  |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_FE\_M\_MONTIMOUT\_REG レジスタ関数モニタタイムアウト

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_NODBGCPUMODE デバッグ対象のCPU動作モードではないためアクセスできない

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

\*reg\_charに指定された<register ID>の内容をmodeで指定された条件によって\*dataに読み込む。

<register ID>は、3.4 レジスタIDの表 3‑27レジスタID一覧を参照

[引数説明]

numは最大32個まで指定できる。

\*dataは読み出したレジスタ値を格納する。

64bitレジスタを読み出した場合は、1回のex\_regreadで下位32bit、上位32bitの順に64bit分のデータを\*dataに格納する。

128bitレジスタを読み出した場合は、1回のex\_regreadで最下位32bit、下位32bit、上位32bit、最上位32bitの順に128bit分のデータを\*dataに格納する。

[動作説明]

ホットプラグインRUN中(接続～強制ブレークするまで)は、mode=EX\_ER\_OFF\_REG\_READをエラー(EX\_SE\_USRPGMRUN)とする。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

G4世代の仮想化支援機能が有効状態で、デバッグ対象ではないCPU動作モード時にレジスタアクセスを行うと、エラー(EX\_SE\_NODBGCPUMODE)を返す。ただし、PC、SP(R3)、FP(R28)、LP(R31)に関しては、デバッガ(CS+、MULTI)がデバッグ情報として必須のため、エラー応答の対象外とする。一度に複数個のレジスタ読出しをする場合は、上記の対象レジスタのチェックをせずにエラー応答する。上記のレジスタは個別に読み出すこと。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・操作対象のPEが停止(初期停止・Cyclic Disable)中である場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

ただし、PC、SP(R3)、FP(R28)、LP(R31)に関しては、デバッガ(CS+、MULTI)がデバッグ情報として必須のため、個別に読み出した場合のみダミーの値(0x00000000)を返してエラー応答しない。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

ただし、PC、SP(R3)、FP(R28)、LP(R31)に関しては、デバッガ(CS+、MULTI)がデバッグ情報として必須のため、個別に読み出した場合のみダミーの値(0x00000000)を返してエラー応答しない。

[デバッガ関係者向け注意事項]

128bitレジスタを読み出す場合は、リードデータ格納用に128bit分のバッファを確保すること。

例として128bitレジスタ(FXU WR0レジスタ)をリードした場合のリードデータ格納順を以下に示す。

WR0

├ W0 (0～31bit) 最下位32bit

├ W1 (32～63bit) 下位32bit

├ W2 (64～95bit) 上位32bit

└ W3 (96～127bit) 最上位32bit

b127 b0

WR0 = 00112233445566778899AABBCCDDEEFF の時、ex\_regreadでWR0をリードする場合

[input]

mode 任意

num 1 (任意だが、本例では1個とする)

\*reg\_char WR0のレジスタ番号 0x0030 → バッファの値「30 00」

\*data (リードデータ格納用に予め128bit分のバッファを確保しておく)

[output] W0 W1 W2 W3

\*data 0xCCDDEEFF, 0x8899AABB, 0x44556677, 0x00112233  
 →バッファの値  
 「FF EE DD CC BB AA 99 88 77 66 55 44 33 22 11 00」

[仮想マシン対応時の注意点(G3世代)]

1. 対象はex\_socunitinfo2のunit\_type=EX\_UNIT\_COREPESELで指定された仮想マシンおよびスレッドのコンテキストのみとする。
2. 仮想マシンコンテキストおよびスレッドコンテキストは、有効ではない番号でもコンテキストは存在するため、アクセス可能となる(CPUコアがサポートしないコンテキストがアクセスできる訳ではない)また、仮想化支援機能に対応しないPEの場合は、ネイティブコンテキストを対象としたアクセスを行う。
3. マシンおよびスレッドの対象となるコンテキストについては、「V850 E3v5アーキテクチャ仕様書」を参照すること。

[仮想マシン対応時の注意点(G4世代)]

1. アクセス対象はI/F呼び出し時のCPU動作モードのコンテキストレジスタとなる。
2. I/F呼び出し時のCPU動作モードがデバッグ対象外の場合はエラー応答し、アクセスしない。  
   ただし、PC、SP(R3)、FP(R28)、LP(R31)は対象外とする。PC、SP(R3)、FP(R28)、LP(R31)は個別にアクセスすること。
3. I/F呼び出し時のCPU動作モードがデバッグ対象の場合、I/F呼び出し時にUM権やSV権であっても、内部的にはHV権でアクセスするため、すべてのレジスタに対してアクセスが可能となる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_regwrite | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

レジスタにデータを書き込む

[Format]

int ex\_regwrite( USHORT mode, USHORT num, USHORT \*red\_char, ULONG \*data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 操作モード | ○ |
| USHORT | num | <I > | 対象レジスタ個数 | ○ |
| USHORT | \*reg\_char | <I > | レジスタID(複数個可) | ○ |
| ULONG | \*data | < O> | アクセスデータ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | EX\_ER\_ON\_REG\_WRITE | ユーザプログラム実行中に呼び出された時、エラー(EX\_SE\_USRPGMRUN)とする。 |
|  | EX\_ER\_OFF\_REG\_WRITE | ユーザプログラム実行中に呼び出された時、ex\_stop処理をしてからレジスタを書き込む。書き込み後再実行する。 |
|  | ~~EX\_M\_DMM\_REG\_WRITE~~ | ~~ユーザプログラム実行中に呼び出された時、一瞬ブレークさせてデータを書き込む。~~ |
|  | 注：RH850ではEX\_M\_DMM\_REG\_WRITEはサポートしない | |
| Num | | |
|  | <0001 - 0020h> | 書き込むレジスタ個数を指定する。 |
|  |  |  |
| \*reg\_char | | |
|  | <register ID> | レジスタ番号(複数個設定可) |
|  |  |  |
| \*data | | |
|  | <00000000 - ffffffffh> |  |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_FE\_M\_MONTIMOUT\_REG レジスタ関数モニタタイムアウト

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_NODBGCPUMODE デバッグ対象のCPU動作モードではないためアクセスできない

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

\*reg\_charに指定された<register ID>の内容をmodeで指定された条件によって\*dataを書き込む。

<register ID>は、3.4 レジスタIDの表 3‑27レジスタID一覧を参照

[引数説明]

numは最大32個まで指定できる。

\*dataはレジスタに書き込む値を格納する。

64bitレジスタに書き込む場合は、1回のex\_regwriteで下位32bit、上位32bitの順に64bit分のデータを\*dataに格納すること。

128bitレジスタに書き込む場合は、1回のex\_regwriteで最下位32bit、下位32bit、上位32bit、最上位32bitの順に128bit分のデータを\*dataに格納すること。

[動作説明]

ホットプラグインRUN中(接続～強制ブレークするまで)は、mode=EX\_ER\_OFF\_REG\_WRITEをエラー(EX\_SE\_USRPGMRUN)とする。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

G4世代の仮想化支援機能が有効状態で、デバッグ対象ではないCPU動作モード時にレジスタアクセスを行うと、エラー(EX\_SE\_NODBGCPUMODE)を返す。ただし、PC、SP(R3)、FP(R28)、LP(R31)に関しては、デバッガ(CS+、MULTI)がデバッグ情報として必須のため、エラー応答の対象外とする。一度に複数個のレジスタ書き込みをする場合は、上記の対象レジスタのチェックをせずにエラー応答する。上記のレジスタは個別に書き込むこと。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・操作対象のPEが停止(初期停止・Cyclic Disable)中である場合、アクセスを行わずエラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[デバッガ関係者向け注意事項]

128bitレジスタに書き込む場合は、ライトデータ格納用バッファに128bit分のデータを配置すること。

ex\_regreadにリードデータ格納例がある。ライトデータ格納順も同様であるため参考とすること。

[仮想マシン対応時の注意点(G3世代)]

1. 対象はex\_socunitinfo2のunit\_type=EX\_UNIT\_COREPESELで指定された仮想マシンおよびスレッドのコンテキストのみとする。
2. 仮想マシンコンテキストおよびスレッドコンテキストは、有効ではない番号でもコンテキストは存在するため、アクセス可能となる(CPUコアがサポートしないコンテキストがアクセスできる訳ではない)  
   また、仮想化支援機能に対応しないPEの場合は、ネイティブコンテキストを対象としたアクセスを行う。
3. マシンおよびスレッドの対象となるコンテキストについては、「V850 E3v5アーキテクチャ仕様書」を参照すること。

[仮想マシン対応時の注意点(G4世代)]

1. アクセス対象はI/F呼び出し時のCPU動作モードのコンテキストレジスタとなる。
2. I/F呼び出し時のCPU動作モードがデバッグ対象外の場合はエラー応答し、アクセスしない。  
   ただし、PC、SP(R3)、FP(R28)、LP(R31)は対象外。PC、SP(R3)、FP(R28)、LP(R31)は個別にアクセスすること。
3. I/F呼び出し時のCPU動作モードがデバッグ対象の場合、I/F呼び出し時にUM権やSV権であっても、内部的にはHV権でアクセスするため、すべてのレジスタに対してアクセスが可能となる。

### イベント条件

ユーザプログラム実行時の各種トリガの条件登録/参照/削除の各関数説明を行う。

1)イベント条件を設定/参照/削除する。

ex\_rh\_evncond

2)イベント統合条件を設定/参照/削除する。

ex\_rh\_evnintgcond

イベント番号

CPUコアイベント番号およびTEU\_CPUイベント番号： 0001h - 3fefh

TEUのスレーブイベント番号： 4000h – 7fdfh

統合イベント番号： 8000h - ffffh

0000hはイベント番号に使用できない

※データアクセスイベントのPE数分の1chをトレースで占有するため、以下の番号は使用できない。

CPUコアイベント番号およびTEU\_CPUイベント番号： 3ff0h - 3fffh

TEUのスレーブイベント番号： 7fe0h – 7fffh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_evncond **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

イベント条件を設定/参照/削除する

[Format]

int ex\_evncond( USHORT func, USHORT no, EXEVCOND \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| USHORT | no | <I > | 単体イベント番号 | -- |
| EXEVCOND | \*cond | <IO> | イベント条件 | -- |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| no | | |
|  | ----- | ----- |
|  |  |  |
| \*cond | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_evncond | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

イベント条件を設定/参照/削除する

[Format]

int ex\_rh\_evncond( USHORT func, USHORT no, EXRHEVCOND \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | no | <I > | 単体イベント番号 | ○ |
| EXRHEVCOND | \*cond | <IO> | イベント条件 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <IO> | アドレス条件の下限値 | ○ |
|  | ULONG | eaddr | <IO> | アドレス条件の上限値 | ○ |
|  | ULONG | addr\_mask | <IO> | アドレス条件のマスク | ○ |
|  | ULONG | data[2] | <IO> | データ条件 | ○ |
|  | ULONG | data\_mask[2] | <IO> | データ条件のマスク | ○ |
|  | USHORT | acc\_size | <IO> | アクセスサイズ | ○ |
|  | USHORT | stat | <IO> | ステータス条件 | ○ |
|  | USHORT | extd | <IO> | 予備 | -- |
|  | USHORT | extd\_mask | <IO> | 予備 | -- |
|  | ULONG | acc\_src | <IO> | アクセス元の指定 | -- |
|  | ULONG | vmcpu\_id | <IO> | 仮想CPU番号 | -- |
|  | ULONG | hwth\_id | <IO> | ハードウェアスレッドの番号 | -- |
|  | ULONG | extra | <IO> | 予備 | -- |
| }EXRHEVCOND | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  | EX\_DEL | 削除 |
|  |  |  |
| no | | |
|  | <0001h - 3fefh> | CPUコアイベント番号およびTEU\_CPUイベント番号 |
|  | <4000h - 7fdfh> | TEUスレーブイベント番号 |
|  |  | を設定する。 |
|  |  |  |
| saddr | | |
|  | <address value> | 下限値のアドレスを指定する。 |
|  |  |  |
| eaddr | | |
|  | <address value> | 上限値のアドレスを指定する。 |
|  |  |  |
| addr\_mask | | |
|  |  | アドレス条件のマスクを指定する。 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| data[2] | | |
|  | <00000000 - ffffffffh> | データ条件を設定する。Don't careの場合は0を指定する |
|  |  |  |
| data\_mask[2] | | |
|  | <00000000 - ffffffffh> | データマスク条件を設定する。Don't careの場合はffffffffhを指定する |
|  |  |  |
| acc\_size | | |
|  | EX\_DATA\_SIZE\_1 | 1バイトアクセス |
|  | EX\_DATA\_SIZE\_2 | 2バイトアクセス |
|  | EX\_DATA\_SIZE\_4 | 4バイトアクセス |
|  | EX\_DATA\_SIZE\_8 | 8バイトアクセス |
|  | EX\_DATA\_SIZE\_12 | 12バイトアクセス |
|  | EX\_DATA\_SIZE\_16 | 16バイトアクセス |
|  | EX\_DATA\_SIZE\_NC | 全アクセスサイズ |
|  |  |  |
| stat | | |
|  | <TEU\_CPUイベント条件> | |
|  | EX\_M1\_STS | 命令フェッチ(実行後) |
|  |  |  |
|  | <TEU\_CPUイベント条件>、<TEUスレーブイベント条件> | |
|  | EX\_RWP\_STS | CPU、スレーブのデータリードライト |
|  | EX\_RP\_STS | CPU、スレーブのデータリード |
|  | EX\_WP\_STS | CPU、スレーブのデータライト |
|  | EX\_RWPND\_STS | CPU、スレーブのデータリードライトで、データが不一致 |
|  | EX\_RPND\_STS | CPU、スレーブのデータリードで、データが不一致 |
|  | EX\_WPND\_STS | CPU、スレーブのデータライトで、データが不一致 |
|  |  |  |
|  | <CPUコアイベント条件>※一部のイベント条件を除き、基本は実行前PCブレークである | |
|  | EX\_RUNB\_STS | CPUの命令系のコアブレークイベント |
|  | EX\_RUNB\_RW\_STS | CPUのアクセス系のデータリードライト |
|  | EX\_RUNB\_R\_STS | CPUのアクセス系のデータリード |
|  | EX\_RUNB\_W\_STS | CPUのアクセス系のデータライト |
|  | EX\_RUNB\_RWND\_STS | CPUのアクセス系のデータリードライトで、データが不一致 |
|  | EX\_RUNB\_RND\_STS | CPUのアクセス系のデータリードで、データが不一致 |
|  | EX\_RUNB\_WND\_STS | CPUのアクセス系のデータライトで、データが不一致 |
|  |  |  |
|  | <アクセスイベントの共通条件> | |
|  | EX\_M\_RANGOUT\_FLG | アドレス範囲外指定のフラグ。他のステータスとORして使う。 |
|  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| acc\_src | | | | | | |
| 【RH850 G3世代MCUの場合】 | | | | | | |
|  | bit0 | TEU\_GRM：DMA、AHB、AXIマスタ、など  TEU\_GVC/TEU\_KVC/TEU\_AXI：DMA  TEU\_EMS：DMA | | | | 0：無効、1：有効 |
|  | bit1 | TEU\_GRM：MAU, AUDR  TEU\_GVC/TEU\_KVC/TEU\_AXI：MAU、AUDR  TEU\_EMS：MAU, AUDR | | | | 0：無効、1：有効 |
|  | bit2 | TEU\_GRM：PE1  TEU\_GVC/TEU\_KVC/TEU\_AXI：外部AHB、AXIマスタ  TEU\_EMS：AHB | | | | 0：無効、1：有効 |
|  | bit3 | TEU\_GRM：PE2  TEU\_GVC/TEU\_KVC/TEU\_AXI：PCU (0固定)  TEU\_EMS：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit4 | TEU\_GRM：PE3  TEU\_GVC/TEU\_KVC/TEU\_AXI：未使用のため0固定  TEU\_EMS：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit5 | TEU\_GRM：PE4  TEU\_GVC/TEU\_KVC/TEU\_AXI：未使用のため0固定  TEU\_EMS：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit6 | TEU\_GRM：PE5  TEU\_GVC/TEU\_KVC/TEU\_AXI：未使用のため0固定  TEU\_EMS：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit7 | TEU\_GRM：PE6  TEU\_GVC/TEU\_KVC/TEU\_AXI：未使用のため0固定  TEU\_EMS：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit8-31 | 未使用 | | | | 0固定 |
|  | ※bit0-7：スレーブ資源により、アクセス先の情報が異なり、<TEUスレーブイベント条件>においては「bit0-7」の指定がなければエラー(EX\_PE\_DATA)を返す。「bit0-7」の指定については妥当チェックを行わないため、ビットの指定に注意ください。 | | | | | |
|  |  |  | | | |  |
| 【RH850 G4世代MCUの場合】 | | | | | | |
|  | bit0 | TEU\_CRM：バス・マスタID0  TEU\_DMA：DMA0  TEU\_LRM：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit1 | TEU\_CRM：バス・マスタID1  TEU\_DMA：DMA1  TEU\_LRM：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit2 | TEU\_CRM：バス・マスタID2  TEU\_DMA：DTS  TEU\_LRM：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | bit3-31(n) | TEU\_CRM：バス・マスタIDn  TEU\_DMA：未使用のため0固定  TEU\_LRM：未使用のため0固定 | | | | 0：無効、1：有効 |
|  | ※1 bit0-31：スレーブ資源により、アクセス先の情報が異なり、また指定について妥当チェックを行わないため、ビットの指定に注意ください。  ※2 バス・マスタIDの割当に関しては各デバイスUMのSPIDおよびBMIDを参照 | | | | | |
|  |  |  | | | |  |
| ※下記は、未使用パラメータのため、0固定。 | | | | | | |
| vmcpu\_id | | | | | | |
|  | | <0 - 000fh> | | 無効 | | |
|  | |  | |  | | |
| hwth\_id | | | | | | |
|  | | | <0 - 00ffh> | | 無効 | |
|  | | |  | |  | |
| extra | | | | | | |
|  | | | <0 - ffffh> | | 無効(予約引数) | |
|  | | |  | |  | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_NONEVNNO 条件未設定のイベント番号を操作した

EX\_PE\_M\_EVNUSED 指定イベントは現在使用されている

EX\_PE\_EVNOVER イベント設定数オーバー

EX\_SE\_TRACEON トレース実行中

EX\_SE\_TIMERON タイマ実行中

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_PE\_INVEVNCOND ハードに設定できないイベント設定を行った

EX\_PE\_SLVEVNNUMUSED 指定イベントは現在違うスレーブ資源で使用されている

EX\_DE\_SELEVTSLAVENOMOUNT 選択されたスレーブ資源が実装されてない、または未サポートである

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_PE\_ACCEVNDATANOSPT　　 アクセスイベント機能でデータ値を指定する設定は未サポート

EX\_PE\_INSTEVNRANGNOSPT 命令実行によるイベント機能で範囲イベント設定は未サポート

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

|  |  |  |
| --- | --- | --- |
| [引数説明] | | |
| no | スレーブ系イベントとスレーブ系以外のイベント番号の指定範囲が下記のように異なるため、イベント条件とnoとの組み合わせが合わない場合は、EX\_PE\_DATAを返す。  1.<TEU\_CPUイベント条件>および<CPUコアイベント条件>の場合、<0001h - 3fefh>  2.<TEUスレーブイベント条件>の場合は、<4000h - 7fdfh>  なお、同じ番号でex\_rh\_evncondを呼ばれた場合は、イベント条件を変えることになるため注意。  例えば、no=0001hでTEU\_CPUイベント条件を設定した後に、同じ番号のno=0001hでCPUコアイベント条件を設定した場合、イベント番号no=0001hにはCPUコアイベント条件のイベントが設定される。同じイベント番号で命令系のイベントからアクセス系のイベントに変更した場合も同様。  また、noに設定した単体イベント番号が現在選択しているPE以外で設定済みの単体イベント番号だった場合は、funcの指定に関わらずエラー(EX\_PE\_M\_EVNUSED)を返し処理は行わない。スレーブイベントの場合も同じく、noに設定した単体イベント番号が違うスレーブ資源の単体イベントとして設定されている場合は、エラー(EX\_PE\_SLVEVNNUMUSED)を返し処理は行わない。 | |
|  |  | |
| stat | noとstatは表 3‑7の組み合わせで設定する必要がある。 | |
| <TEU\_CPUイベント条件> | |
| 実行後のブレークとなり、PE単位でイベント設定可能。 | |
|  | |
| <CPUコアイベント条件> | |
| コアイベントを利用した実行系のイベント(EX\_RUNB\_STS)とアクセス系のイベント(EX\_RUNB\_XXXX\_STS)を指定するときに用いる。一部のイベント条件を除き実行前にブレークし、PE単位で設定可能。  RH850 G4世代MCUの場合は、G3KコアをもつICUMも含めて、EX\_RUNB\_RWND\_STS、EX\_RUNB\_RND\_STS、  EX\_RUNB\_WND\_STSは選択できない。またdata\_mask[1:0]に0xffffffff以外を指定した状態で、EX\_RUNB\_RW\_STS、EX\_RUNB\_R\_STS、EX\_RUNB\_RW\_STSも選択できない。選択した場合はエラー(EX\_PE\_ACCEVNDATANOSPT)を返す。 | |
|  | |
| <TEUスレーブイベント条件> | |
| TEUスレーブイベントとして、命令系のイベントは指定できない。指定した場合、エラー(EX\_PE\_DATA)を返す。アクセス系のイベントについては、<TEU\_CPUイベント条件>と同じstatを利用する。コアの外部資源を利用するため、PE単位で設定できない。TEUスレーブイベントもTEU\_CPUイベントと同じく実行後のブレークになる。  スレーブイベント資源の類を選択するには、ex\_socunitinfo2のEX\_UNIT\_SLAVEIDSELを利用すること。デバイスによって搭載するスレーブイベント資源が異なる。デバイス別に搭載するスレーブ資源については表 3‑9を参照すること。EX\_UNIT\_SLAVEIDSELより選択されたスレーブ資源がTEUに搭載しない場合、エラー(EX\_DE\_SELEVTSLAVENOMOUNT)を返す。  スレーブイベント資源としてTEU\_CRMを選択した場合、クラスタ番号の指定はex\_socunitinfo2のEX\_UNIT\_CLUSTERSELを利用すること。 | |
|  | |
| <アクセスイベントの共通条件> | |
| <CPUコアイベント条件>以外のときは、stat条件とEX\_M\_RANGOUT\_FLGをORして指定すると範囲外のイベントを設定することが可能。<CPUコアイベント条件>とORした場合、エラー(EX\_PE\_DATA)を返す。 | |
|  |  |
| ※statとデータサイズの組み合わせについては、表 3‑8を参照。 | |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| addr\_mask | アクセス系イベント検出器の場合、addr\_maskはアドレス条件を有効無効にするスイッチとして用いる。0xFFFFFFFFを指定するとアドレス条件が無効になる。データのみの条件にしたい場合、アドレス条件を無効にすると検出器は1個で済む。 | | |
|  |  | | |
| data[2] | acc\_sizeの設定によって、以下のように下位～上位の順に(ワード単位またはバイト単位で)、データ条件を設定する。 | | |
| acc\_size= EX\_DATA\_SIZE\_1 | data[0] = 000000XXh; data[1] = 00000000h; | |
| acc\_size= EX\_DATA\_SIZE\_2 | data[0] = 0000XXXXh; data[1] = 00000000h; | |
| acc\_size= EX\_DATA\_SIZE\_4 | data[0] = XXXXXXXXh; data[1] = 00000000h; | |
| acc\_size= EX\_DATA\_SIZE\_8 | data[0] = XXXXXXXXh; data[1] = XXXXXXXXh; | |
| acc\_size= EX\_DATA\_SIZE\_12 | data[0] = 00000000h; data[1] = 00000000h; | |
| acc\_size= EX\_DATA\_SIZE\_16 | data[0] = 00000000h; data[1] = 00000000h; | |
| acc\_size= EX\_DATA\_SIZE\_NC | data[0] = 00000000h; data[1] = 00000000h; | |
| XX：設定値、0：設定不要 | | |
| IEハード上のイベント検出器は4種類あり、種類によりデータサイズの指定は異なるので注意。検出器の種類とデータ条件の設定関係を表 3‑8に示す。EX\_DATA\_SIZE\_12、EX\_DATA\_SIZE\_16、EX\_DATA\_SIZE\_NCの場合、データ比較はできないためデータ条件を上記のように固定値を設定すること。acc\_sizeを超えるdataが入力された場合はエラー(EX\_PE\_DATA)を返す。 | | |
|  |  | | |
| data\_mask[2] | acc\_sizeの設定によって、以下のように下位～上位の順に(ワード単位またはバイト単位で)、データのマスク条件を設定する。 | | |
| acc\_size= EX\_DATA\_SIZE\_1 | | data\_mask[0] = FFFFFFXXh; data\_mask[1] = FFFFFFFFh; |
| acc\_size= EX\_DATA\_SIZE\_2 | | data\_mask[0] = FFFFXXXXh; data\_mask[1] = FFFFFFFFh; |
| acc\_size= EX\_DATA\_SIZE\_4 | | data\_mask[0] = XXXXXXXXh; data\_mask[1] = FFFFFFFFh; |
| acc\_size= EX\_DATA\_SIZE\_8 | | data\_mask[0] = XXXXXXXXh; data\_mask[1] = XXXXXXXXh; |
| acc\_size= EX\_DATA\_SIZE\_12 | | data\_mask[0] = FFFFFFFFh; data\_mask[1] = FFFFFFFFh; |
| acc\_size= EX\_DATA\_SIZE\_16 | | data\_mask[0] = FFFFFFFFh; data\_mask[1] = FFFFFFFFh; |
| acc\_size= EX\_DATA\_SIZE\_NC | | data\_mask[0] = FFFFFFFFh; data\_mask[1] = FFFFFFFFh; |
| XX：設定値、FF：設定不要 | | |
| マスク条件は、1が立っているビットに対してデータ条件の該当ビットが1でも0でも有効となる。例えば、データ条件でdata：0x50、data\_mask：0x0Fと設定した場合、0x50～0x5Fが一致条件となる。  EX\_DATA\_SIZE\_12、EX\_DATA\_SIZE\_16、EX\_DATA\_SIZE\_NCの場合、データ比較はできない。データ条件をマスクするように上記のような設定を行う。 | | |
|  |  | | |
| acc\_size | statでデータアクセスイベントを設定したときにイベント検出を行うアクセスサイズを設定する。   * statでコアのアクセスイベントを指定して、EX\_DATA\_SIZE\_12~~、EX\_DATA\_SIZE\_16~~を指定した場合は、エラー(EX\_PE\_DATA)を返す。 * statでコアのアクセスイベントを指定して、EX\_DATA\_SIZE\_16を指定した場合は、選択コアがG4MHコアの場合のみ受け付け、その他のコアではエラー(EX\_PE\_DATA)を返す。 * statでCPU、スレーブでアクセスイベントを指定したとき、設定先によって指定可能なアクセスサイズは異なり、設定できないアクセスサイズが指定された場合は、エラー(EX\_PE\_DATA)を返す。 | | |
|  |  | | |
| acc\_src | <TEU\_CPUイベント条件>および<CPUコアイベント条件>の場合、acc\_srcの指定が不要になるため、acc\_srcの指定があってもエラーを返さない。acc\_srcの指定は不要。 | | |
|  | <TEUスレーブイベント条件>の場合、アクセス元の指定が必須になるため、acc\_srcの指定がない場合、エラー(EX\_PE\_DATA)を返す。 | | |

|  |
| --- |
| [動作説明] |
|  |
| 設定[EX\_SET] |
| 単体イベント番号で示したイベント条件(アドレス、データ、アクセスサイズ、ステータス、外部データ)を設定する。IEハード上のイベント検出器の個数を表 3‑8に示す。この数をオーバーする設定があった場合、エラー(EX\_PE\_EVNOVER)を返す。アドレスの範囲指定の場合はイベント検出器を2つ使用する。アクセス系イベントのデータ指定で64bitの場合もイベント検出器を2つ使用する。なお、96bit、128bitの場合は、使用するイベント検出器は1つのみ。 |
| 参照[EX\_REF] |
| 単体イベント番号で示したイベント条件を参照する。未設定の単体イベント番号を指定するとエラー(EX\_SE\_NONEVNNO)を返す。 |
| 削除[EX\_DEL] |
| 単体イベント番号で示したイベント条件を削除する。未設定の単体イベント番号を指定しても、エラーとしない。アクションに設定されている単体イベント番号の削除を行った場合、アクションも消えるので注意すること。 |
| 参照/削除の場合、イベント条件は不定でよい。  ユーザプログラム実行中でも追加設定は可能。ただし、トレース関係のイベントに設定済みのイベントは、トレース実行中の変更は不可能。タイマの開始/終了イベント、および、パフォーマンス計測機能の開始/終了イベントに設定済みのイベントを、ユーザプログラム実行中に変更/削除すると、エラー(EX\_SE\_USRPGMRUN)を返す。  CPUのコアイベントをユーザプログラム実行中に変更する場合は、強制ブレークで止めて設定し再実行させる(ブレーク期間はホストPCの時間間隔)。  ホットプラグインRUN中(接続～強制ブレークするまで)は、イベント条件の種別に関わらず、EX\_SE\_USRPGMRUNエラーとする。(次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中はCPUコアイベント条件の設定のみEX\_SE\_USRPGMRUNエラーとし、その他のイベント条件は設定可能とする。) |
| RH850 G4世代MCUの場合、TEU\_CPUの命令系範囲イベントを使用できないMCUがある。該当するMCUを使用していた場合に、TEU\_CPUの命令系範囲イベントを設定するとエラー応答(EX\_PE\_INSTEVNRANGNOSPT)する。 |
| E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。 |

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・操作対象のPEが停止(初期停止・Cyclic Disable)中で、CPUコア(実行前)イベントを設定/削除した場合は、エラー応答(EX\_SE\_CPUSTOP\_NOSPRT)して要求を受け付けない。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、CPUコア(実行前)イベントだけなくTEUイベント(実行後)も含め設定/削除した場合は、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して要求を受け付けない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| <TEU\_CPUイベント条件>もしくは<TEUスレーブイベント条件>の設定を行うときは、ex\_rh\_evncond呼び出す前に必ず、ex\_socunitinfo2のEX\_UNIT\_SLAVEIDSELより、設定イベントの類(<TEU\_CPUイベント>もしくは<TEUスレーブイベント>)の選択が必要になる。ex\_socunitinfo2のEX\_UNIT\_SLAVEIDSELで指定するunit\_info[0]が0x0000の場合、<TEU\_CPUイベント条件>が有効となり、そうでない場合は、<TEUスレーブイベント条件>が有効になる。なお、ex\_socunitinfo2のEX\_UNIT\_SLAVEIDSELで指定するunit\_info[0]が0x0000以外の場合、EX\_UNIT\_SLAVEIDSELより選択されたスレーブ資源が、TEUに搭載しない場合エラー(EX\_DE\_SELEVTSLAVENOMOUNT)を返す。  <TEU\_CRMイベント条件>の設定を行うときは、ex\_rh\_evncondを呼び出す前に必ず、ex\_socunitinfo2のEX\_UNIT\_CLUSTERSELにより、クラスタ番号の設定が必要になる。 |
| IEハード上のイベント検出器の個数を表 3‑8に示す。検出機種によって検出器の個数が異なる。この数をオーバーする設定があった場合、エラー(EX\_PE\_EVNOVER)を返す。  基本的には、イベントの設定関数を呼ばれた時点で、空いていれば小さい番号のイベントを使う。ポイント1個設定後に範囲設定をすると、歯抜けの空きがでたり、イベント削除によっても歯抜けの空きがでたりする。なお、歯抜けで飛び飛びに2個イベントがあるときに、範囲イベントの設定もしくは64ビットのアクセスイベントを設定した場合は、既にハードに登録してあるイベントを付け替えて、飛び飛びに2個あるイベントを連続2個にする。  TEU系のイベントの設定はユーザプログラム実行中でも可能である。ただし、範囲イベントの設定もしくは64ビットのアクセスイベントの設定があるときのイベント番号の付け替えは、ユーザプログラム実行中もしくはトレース実行中のであれば行わない。このときは、ユーザプログラム実行中(EX\_SE\_USRPGMRUN)もしくはトレース実行中(EX\_SE\_TRACEON)のエラーを返す。なお、イベント番号の付け替えは、<CPUコアイベント条件>にて64ビットのアクセスイベントがあった場合でも上記の理由で付け替え処理を行う。  また、デバッガ側から「データアクセスのトレース情報を取得する」要求があるときは、TEUのアクセス系イベントはトレース側で利用されるため、チャネル数が1つ減る。なお、EXEC内部でデータアクセスのトレース情報を取得するため、トレース側が以下の番号を利用するので以下の番号が使用できない。以下の番号を使用した場合、エラー(EX\_PE\_DATA)を返す。  CPUイベント番号：3ff0h - 3fffh  スレーブ用のイベント番号：7fe0h – 7fffh |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 【RH850 G3世代MCUの場合】  TEUのスレーブイベントで利用する「acc\_src」引数の意味は、スレーブ資源ごとで下記のように異なる。   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | bit0 | bit1 | bit2 | bit3 | bit4 | bit5 | bit6 | bit7 | bit8-31 | | TEU\_GRM | DMA, AHB, AXIマスタなど | MAU, AUDR | PE1 | PE2 | PE3 | PE4 | PE5 | PE6 | 0固定 | | TEU\_GVC  TEU\_KVC  TEU\_AXI | DMA | MAU, AUDR | AHB, AXI外部マスタ | PCU | 0固定 | 0固定 | 0固定 | 0固定 | 0固定 | | TEU\_EMS | DMA | MAU, AUDR | AHBマスタ | 0固定 | 0固定 | 0固定 | 0固定 | 0固定 | 0固定 |   ※bit0-7：スレーブ資源により、アクセス先の情報が異なり、また「bit0-7」の指定の妥当チェックを行わないため、注意ください。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 【RH850 G4世代MCUの場合】  TEUのスレーブイベントで利用する「acc\_src」引数の意味は、スレーブ資源ごとで下記のように異なる。   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | bit0 | bit1 | bit2 | bit3-31(n) | | TEU\_CRM | バス・マスタID0 | バス・マスタID1 | バス・マスタID2 | バス・マスタIDn | | TEU\_DMA | DMA0 | DMA1 | DTS | 0固定 | | TEU\_LRM | 0固定 | 0固定 | 0固定 | 0固定 |   ※1 bit0-31：スレーブ資源により、アクセス先の情報が異なり、また指定の妥当チェックを行わないため、注意ください。  ※2 バス・マスタIDの割当に関しては各デバイスUMのSPIDおよびBMIDを参照 |

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| 96bit, 128bitのアクセスイベントに関する注意事項：  96bit, 128bitのイベント検出ではデータ値の比較はできない。  96bit, 128bitアクセスが発生するのはLocalRAM(TCM)へのライトのみ。  LocalRAM(TCM)からのリードは最大64bitアクセス。  LocalRAM(TCM)以外のリード/ライトは最大64bitアクセス。  なお、コアイベントのアクセスイベントも、最大64bitまでアクセス可能となる。  アクセスサイズを全アクセスサイズとした場合、データの比較はできない。  statにEX\_xxPND\_STS、およびEX\_RUNB\_xxND\_STSのデータ不一致のアクセスイベントを設定する場合、指定  アクセスサイズを全bitマスク設定にできない。 |
| Phase2以降、PFC1Bと共にPFC1AのTEU\_CPUイベントの本数を4本から8本に変更する。詳細は表 3‑8を参照すること。 |
| <CPUコアイベント条件>のstatを条件としたときは一部を除いて命令実行前でブレークが発生する。  一部実行後ブレークとなるケースは以下の2点になる。  ① リードアクセス(EX\_RUNB\_RW\_STS、EX\_RUNB\_R\_STS、EX\_RUNB\_RWND\_STS、EX\_RUNB\_RND\_STS)の指定で、データ条件を指定したとき。  ② ライトアクセス(EX\_RUNB\_RW\_STS、EX\_RUNB\_W\_STS、EX\_RUNB\_RWND\_STS、EX\_RUNB\_WND\_STS)の指定で、リードモディファイライト系の命令のライトアクセスが検出されたとき。 |
| <CPUコアイベント条件>を使用したイベントの場合、実行開始アドレスがイベント成立条件の場合、そのアドレスを1ステップ実行してから通常のRUN処理を行う。<TEU\_CPUイベント条件>を使用したイベントの場合、実行開始アドレスがイベント成立条件の場合でも、そのアドレスを1ステップ実行せず通常のRUN処理を行う。1ステップ実行を行うときは一旦すべてのブレークイベントを削除し、1ステップ実行が完了してから改めて削除したブレークイベントの条件を設定する。このため、1ステップ実行中はブレークイベントが検出できない。 |
| <TEU\_GRM/TEU\_CRM>, <TEU\_GVC/TEU\_KVC/TEU\_AXI/TEU\_DMA>, <TEU\_EMS>, <TEU\_LRM>を用いたスレーブイベント機能においては、G3K, G3K-Hコアの場合は64bitのデータアクセスイベント検出はできない。この制限はG3Kを搭載するマルチコア(PFC1B)で、PE3を用いたスレーブイベントを利用した場合に該当する。また、G3Kを搭載するシングルコアのF1L, R1LデバイスおよびG3K-Hを搭載するシングルコアのF1Kにも該当する。 |
| [仮想マシン対応時の注意点(G3世代)]  イベント機能において、仮想マシン対応時に設定が必要な項目はない。ex\_rh\_evncondを用いて設定されるイベント(TEUおよびコアイベントの命令系イベントとデータアクセスイベントを含む)は品種の区別をせず、ネイティブマシンと仮想マシンのVM#0およびハードウェアスレッドのHT#0へ設定する。  なお、仮想マシン対応においてVM#0とHT#0の設定を行っても、イベントの総数やその他の仮想マシン対応以外の動作に影響はない。 |
| [仮想マシン対応時の注意点(G4世代)]  CPU動作モードごとにイベント検出有無の指定が可能となる。イベント検出有無の指定はデバッグ対象の指定(ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESEL)と連動させる。TEU\_CPUイベントとコアイベントの両方に対して設定を行い、TEUスレーブイベントは設定の対象外となる。  また、デバッグ対象となるCPU動作モードの指定(ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESEL)は、ex\_socunitinfo2のunit\_type=EX\_UNIT\_COREPESELで指定されたコア毎に管理する。イベント設定と対象コアの関係は、ex\_rh\_evncond呼び出し時の選択コア番号とイベントの設定内容を紐付けする。一方、イベント設定とCPU動作モードの関係は、最後に指定されたデバッグ対象のCPU動作モードをイベントに紐付ける。 |

|  |
| --- |
| [マルチコア時の注意点] |
| ① noに設定した単体イベント番号が現在選択しているPE以外で設定済みの単体イベント番号だった場合は、funcの指定に関わらずエラー(EX\_PE\_M\_EVNUSED)を返し処理は行わない。 |
| ② ex\_getenvでは、イベントの個数を返さない。 上位モジュールでのイベント個数の管理については、EX\_PE\_EVNOVERが返るまでイベント設定が可能であることとする。 |
| ③ <TEU\_GRM/TEU\_CRM>, <TEU\_GVC/TEU\_KVC/TEU\_AXI/TEU\_DMA>, <TEU\_EMS>, <TEU\_LRM>を用いたスレーブイベント機能においては、G3Kコアの場合は64bitのデータアクセスイベント検出はできない。この制限はG3Kを搭載するマルチコア(PFC1B)で、PE3を用いたスレーブイベントを利用した場合に該当する。 |

表 3‑7 G3世代MCUのstat条件とイベント条件の対応

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stat条件  イベント条件 | TEUイベント | | | CPUコアイベント | |
| <TEU\_CPU条件> | | <TEUスレーブイベント条件> |
| EX\_M1\_STS | EX\_RWP\_STS,  EX\_RP\_STS,  EX\_WP\_STS,  EX\_RWPND\_STS,  EX\_RPND\_STS,  EX\_WPND\_STS, | EX\_RWP\_STS,  EX\_RP\_STS,  EX\_WP\_STS,  EX\_RWPND\_STS,  EX\_RPND\_STS,  EX\_WPND\_STS | EX\_RUNB\_STS | EX\_RUNB\_RW\_STS  EX\_RUNB\_R\_STS  EX\_RUNB\_W\_STS  EX\_RUNB\_RWND\_STS  EX\_RUNB\_RND\_STS  EX\_RUNB\_WND\_STS |
| no | 0001h-3fefh | 0001h-3fefh | 4000h-7fdfh | 0001h-3fefh | 0001h-3fefh |
| saddr | ○ | ○ | ○ | ○※1 | ○※1 |
| eaddr | ○ | ○ | ○ | ○※1 | ○※1 |
| addr\_mask | 0x00000000 | 0x00000000  または  0xffffffff | 0x00000000  または  0xffffffff | ○ | ○ |
| data[2] | × | ○ | ○ | × | ○ |
| data\_mask[2] | × | ○ | ○ | × | ○ |
| acc\_size | × | ○ | ○ | × | ○ |
| stat | ○※2 | ○※2 | ○※2 |  |  |
| ~~extd~~ | ~~○~~ | ~~○~~ | ~~○~~ | ~~×~~ | ~~×~~ |
| ~~extd\_mask~~ | ~~○~~ | ~~○~~ | ~~○~~ | ~~×~~ | ~~×~~ |
| acc\_src | × | × | ○ | × | × |
| vmcpu\_id | × | × | × | × | × |
| hwth\_id | × | × | × | × | × |
| extra | × | × | × | × | × |
|  | | | | | |
| ○：設定可能、×：設定不可(Don’t Care)、数値：必須設定値  ※1：ポイントアドレス指定のみ  ※2：statへのEX\_M\_RANGOUT\_FLGのOR指定可能もしくはOR指定必須 | | | | | |

表 3‑8 G3世代MCUのacc\_size、statとイベント検出器の個数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 検出器の個数 | | | | |
| acc\_size  stat、スレーブ資源 | EX\_DATA\_SIZE\_1  EX\_DATA\_SIZE\_2  EX\_DATA\_SIZE\_4  EX\_DATA\_SIZE\_NC | EX\_DATA\_SIZE\_8 | EX\_DATA\_SIZE\_12 | EX\_DATA\_SIZE\_16 | Don’t Care |
| EX\_RWP\_STS  EX\_RP\_STS  EX\_WP\_STS  EX\_RWPND\_STS  EX\_RPND\_STS  EX\_WPND\_STS | P⇒8※2 | P⇒4※2 | P⇒8※2 | P⇒8※2 | - |
| R⇒4※2 | R⇒4※2 | R⇒4※2 | R⇒4※2 | - |
| EX\_M1\_STS | - | - | - | - | P⇒8 |
| - | - | - | - | R⇒4 |
| EX\_RUNB\_STS | - | - | - | - | P⇒12※1 |
| - | - | - | - | R⇒- |
| EX\_RUNB\_RW\_STS  EX\_RUNB\_R\_STS  EX\_RUNB\_W\_STS  EX\_RUNB\_RWND\_STS  EX\_RUNB\_RND\_STS  EX\_RUNB\_WND\_STS | P⇒4※1 | P⇒4※1 | - | - | - |
| R⇒- | R⇒- | - | - | - |
| TEU\_TCM | P⇒4※2 | P⇒2※2 | P⇒4※2 | P⇒4※2 | - |
| R⇒2※2 | R⇒2※2 | R⇒2※2 | R⇒2※2 | - |
| TEU\_GRM | P⇒4※2 | P⇒2※2 | - | - | - |
| R⇒2※2 | R⇒2※2 | - | - | - |
| TEU\_GIO | P⇒4※2 | - | - | - | - |
| R⇒2※2 | - | - | - | - |
| TEU\_GVC/TEU\_KVC/  TEU\_AXI | P⇒4※2 | P⇒2※2 | - | - | - |
| R⇒2※2 | R⇒2※2 | - | - | - |
| TEU\_EMS | P⇒4※2 | - | - | - | - |
| R⇒2※2 | - | - | - | - |
| P：ポイント指定  R：範囲指定  -：設定不可  Don't Care：acc\_size設定不要(命令実行系のイベント)  ※1：EX\_RUNB\_STSの12個の内、4個はEX\_RUNB\_XXX\_STSと兼用。従って、EX\_RUNB\_STSのイベント個数が8を超えた場合、EX\_RUNB\_XXX\_STSの設定可能なイベントの個数が減る。逆に、EX\_RUNB\_XXX\_STSイベントとして4個が設定済みの場合、EX\_RUNB\_STSイベントの最大個数は8個となる。  ※2：全トレースでかつデータアクセストレースを使用する場合は、使用できるアクセスイベントのチャネル数が1つ減る。 | | | | | |

表 3‑9 G3世代MCUのデバイス別に搭載するTEUスレーブイベント資源

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| コア種別 | TEU\_CPU資源 | TEUスレーブ資源 | | | | | |
| TEU\_TCM | TEU\_GIO | TEU\_GRM | TEU\_GVC | TEU\_KVC | TEU\_AXI | TEU\_EMS |
| G3M0.52 | △ | △ | △ | ｘ | ｘ | ｘ | ｘ |
| G3M(0.52以外) | ｘ | ｘ | ○ | ○ | ｘ | ｘ | ｘ |
| G3M-H | ｘ | ｘ | ○ | ｘ | ｘ | ○ | ｘ |
| G3K | ｘ | ｘ | ｘ | ｘ | ｘ | ｘ | ○ |
| G3K-H | ｘ | ｘ | ｘ | ｘ | ○ | ｘ | × |

○：搭載のため、設定可能

ｘ：未搭載のため、設定不可

△：搭載されているが、対応してない

表 3‑10 G4世代MCUのstat条件とイベント条件の対応

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stat条件  イベント条件 | TEUイベント | | | CPUコアイベント | |
| <TEU\_CPU条件> | | <TEUスレーブイベント条件> |
| EX\_M1\_STS  ※5, 6 | EX\_RWP\_STS,  EX\_RP\_STS,  EX\_WP\_STS,  EX\_RWPND\_STS,  EX\_RPND\_STS,  EX\_WPND\_STS, | EX\_RWP\_STS,  EX\_RP\_STS,  EX\_WP\_STS,  EX\_RWPND\_STS,  EX\_RPND\_STS,  EX\_WPND\_STS | EX\_RUNB\_STS | EX\_RUNB\_RW\_STS  EX\_RUNB\_R\_STS  EX\_RUNB\_W\_STS  ※2 |
| no | 0001h-3fefh | 0001h-3fefh | 4000h-7fdfh | 0001h-3fefh | 0001h-3fefh |
| saddr | ○ | ○ | ○ | ○※1 | ○※1 |
| eaddr | ○ | ○ | ○ | ○※1 | ○※1 |
| addr\_mask | 0x00000000 | 0x00000000  または  0xffffffff | 0x00000000  または  0xffffffff | ○ | ○ |
| data[2] | × | ○ | ○ | × | × ※2, 3 |
| data\_mask[2] | × | ○ | ○ | × | × ※2, 3 |
| acc\_size | × | ○ | ○※4 | × | ○ |
| stat | ○※3 | ○※3 | ○※3 |  |  |
| ~~extd~~ | ~~○~~ | ~~○~~ | ~~○~~ | ~~×~~ | ~~×~~ |
| ~~extd\_mask~~ | ~~○~~ | ~~○~~ | ~~○~~ | ~~×~~ | ~~×~~ |
| acc\_src | × | × |  | × | × |
| vmcpu\_id | × | × | × | × | × |
| hwth\_id | × | × | × | × | × |
| extra | × | × | × | × | × |
| ○：設定可能、×：設定不可(Don’t Care)、数値：必須設定値  ※1：ポイントアドレス指定のみ  ※2：G4世代MCU内のG4MHコアのコアイベントにはアクセスデータ値を比較する機能がデバイスとして存在しない。G4世代MCU内にG3コアがマルチコアの一部として含まれる場合、G3コアのみアクセスデータ比較が可能である。しかし、G4世代MCU内の機能統一性を優先するためすべてのコアに対して、stat条件にEX\_RUNB\_RWND\_STS、EX\_RUNB\_RND\_STS、EX\_RUNB\_WND\_STSは指定できないこととする。またstat条件にEX\_RUNB\_RW\_STS、EX\_RUNB\_W\_STS、EX\_RUNB\_R\_STSを指定する場合も、data\_mask[1:0]は必ず0xffffffffを指定すること。  ※3：statへのEX\_M\_RANGOUT\_FLGのOR指定可能もしくはOR指定必須  ※4：スレーブ資源がTEU\_CRM, TEU\_LRMの場合、EX\_DATA\_SIZE\_4とEX\_DATA\_SIZE\_NCのみ対応。  スレーブ資源がTEU\_DMAの場合、EX\_DATA\_SIZE\_1, EX\_DATA\_SIZE\_2, EX\_DATA\_SIZE\_4, EX\_DATA\_SIZE\_8, EX\_DATA\_SIZE\_NCに対応。  ※5：G4世代では、TEU\_CPUの命令系範囲イベントに不具合があるデバイスがあり、不具合が該当するデバイス選択時にTEU\_CPUの命令形範囲イベントを指定した場合、エラー応答する(EX\_PE\_INSTEVNRANGNOSPT)。なお、不具合の該当有無はex\_rh\_getenvで参照すること。  ※6：G4世代MCU内にG3コアがマルチコアの一部として含まれている場合、G3コアに対してTEU\_CPUの命令系範囲イベントは設定可能であるが、G4世代MCU内の機能統一性を優先するためEXEC側で全コア設定不可とする。 | | | | | |

表 3‑11 G4世代MCUのacc\_size、statとイベント検出器の個数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 検出器の個数 | | | | |
| acc\_size  stat、スレーブ資源 | EX\_DATA\_SIZE\_1  EX\_DATA\_SIZE\_2  EX\_DATA\_SIZE\_4  EX\_DATA\_SIZE\_NC | EX\_DATA\_SIZE\_8 | EX\_DATA\_SIZE\_12 | EX\_DATA\_SIZE\_16 | Don’t Care |
| EX\_RWP\_STS  EX\_RP\_STS  EX\_WP\_STS  EX\_RWPND\_STS  EX\_RPND\_STS  EX\_WPND\_STS | P⇒8※2 | P⇒4※2 | - | P⇒8※2 | - |
| R⇒4※2 | R⇒4※2 | - | R⇒4※2 | - |
| EX\_M1\_STS | - | - | - | - | P⇒8 |
| - | - | - | - | R⇒4 |
| EX\_RUNB\_STS | - | - | - | - | P⇒12※1 |
| - | - | - | - | R⇒- |
| EX\_RUNB\_RW\_STS  EX\_RUNB\_R\_STS  EX\_RUNB\_W\_STS | P⇒4※1 | P⇒4※1 | - | P⇒4※1※3 | - |
| R⇒- | R⇒- | - | R⇒- | - |
| TEU\_CRM | P⇒4※2※4 | - | - | - | - |
| R⇒2※2※4 | - | - | - | - |
| TEU\_DMA | P⇒4※2 | P⇒2※2 | - | - | - |
| R⇒2※2 | R⇒1※2 | - | - | - |
| TEU\_LRM | P⇒4※2※4 | - | - | - | - |
| R⇒2※2※4 | - | - | - | - |
| P：ポイント指定  R：範囲指定  -：設定不可  Don’t Care：acc\_size設定不要(命令実行系のイベント)  ※1：EX\_RUNB\_STSの12個の内、4個はEX\_RUNB\_XXX\_STSと兼用。従って、EX\_RUNB\_STSのイベント個数が8を超えた場合、EX\_RUNB\_XXX\_STSの設定可能なイベントの個数が減る。逆に、EX\_RUNB\_XXX\_STSイベントとして4個が設定済みの場合、EX\_RUNB\_STSイベントの最大個数は8個となる。  ※2：全トレースでかつデータアクセストレースを使用する場合は、使用できるアクセスイベントのチャネル数が1つ減る。  ※3：EX\_RUNB\_XXX\_STSイベントのEX\_DATA\_SIZE\_16指定は、選択コアがG4MHコアのみ可能。  　　 G4世代MCU内のG3コアで設定した場合、エラーとする。G4世代MCU内で設定できるパラメータがコア毎に異なることになるが、G4MHコアのクアッドワードアクセス新命令に対応するためサポートする。  ※4：EX\_DATA\_SIZE\_4とEX\_DATA\_SIZE\_NCのみ対応 | | | | | |

表 3‑12 G4世代MCUのデバイス別に搭載するTEUスレーブイベント資源

|  |  |  |  |
| --- | --- | --- | --- |
| コア種別 | TEUスレーブ資源 | | |
| TEU\_DMA | TEU\_LRM | TEU\_CRM |
| G4MH | ○ | ○ | ○ |

○：搭載のため、設定可能

ｘ：未搭載のため、設定不可

△：搭載されているが、対応してない

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_evnintgcond **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

統合イベント条件(シーケンシャルイベント条件)を設定/参照/削除する

[Format]

int ex\_evnintgcond( USHORT func, USHORT seq\_no, EXINTGCOND \*intg )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| USHORT | seq\_no | <I > | 統合イベント番号 | -- |
| EXINTGCOND | \*intg | <IO> | 統合イベント条件 | -- |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| seq\_no | | |
|  | ----- | ----- |
|  |  |  |
| \*intg | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

1. サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_evnintgcond | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

統合イベント条件(シーケンシャルイベント条件)を設定/参照/削除する

[Format]

int ex\_rh\_evnintgcond( USHORT func, USHORT seq\_no, EXRHINTGCOND \*intg )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | seq\_no | <I > | 統合イベント番号 | ○ |
| EXRHINTGCOND | \*intg | <IO> | 統合イベント条件 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | \*event1 | <IO> | 1段目の単体イベント番号(複数個可) | ○ |
|  | USHORT | \*event2 | <IO> | 2段目の単体イベント番号(複数個可) | ○ |
|  | USHORT | \*event3 | <IO> | 3段目の単体イベント番号(複数個可) | ○ |
|  | USHORT | \*event4 | <IO> | 4段目の単体イベント番号(複数個可) | ○ |
|  | USHORT | \*del\_ev | <IO> | ディゼーブルの単体イベント番号(複数個可) | ○ |
|  | USHORT | seq\_cond | <IO> | シーケンシャルイベント・累積ANDイベントを区別するフラグ | ○ |
|  | ULONG | pasdly\_cnt | <IO> | パスカウント値・ディレイカウント値 | ○ |
| }EXRHINTGCOND | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  | EX\_DEL | 削除 |
|  |  |  |
| seq\_no | | |
|  | <8000h – ffffh> | 統合イベント番号を設定する |
|  |  |  |
| \*event1 | | |
|  | <0001h – 3fefh> | CPUコアイベントおよびTEU\_CPUイベント番号を設定する |
|  | <4000h – 7fdfh> | TEUスレーブイベント番号を設定する |
|  |  |  |
| \*event2 | | |
|  | <0001h – 3fefh> | CPUコアイベントおよびTEU\_CPUイベント番号を設定する |
|  | <4000h – 7fdfh> | TEUスレーブイベント番号を設定する |
|  |  |  |
| \*event3 | | |
|  | <0001h – 3fefh> | CPUコアイベントおよびTEU\_CPUイベント番号を設定する |
|  | <4000h – 7fdfh> | TEUスレーブイベント番号を設定する |
|  |  |  |
| \*event4 | | |
|  | <0001h – 3fefh> | CPUコアイベントおよびTEU\_CPUイベント番号を設定する |
|  | <4000h – 7fdfh> | TEUスレーブイベント番号を設定する |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \*del\_ev | | | | |
|  | | <0001h – 3fefh> | CPUコアイベントおよびTEU\_CPUイベント番号を設定する | |
|  | | <4000h – 7fdfh> | TEUスレーブイベント番号を設定する | |
|  | | 0000hが終端。\*event1-\*event4、\*del\_evのメンバが示すバッファの先頭に0000hが設定されていればイベント条件が設定されていないとみなす | | |
|  | |  | | |
| seq\_cond | | | | |
|  | bit0 | シーケンシャルイベントを利用 | | 0：無効、1：有効 |
|  | bit1 | 累積ANDイベントを利用 | | 0：無効、1：有効 |
|  | bit2 | パスカウントを利用 | | 0：無効、1：有効 |
|  | bit3 | ディレイカウントを利用 | | 0：無効、1：有効 |
|  | bit4-15 | 未使用 | | 0固定 |
|  | | | | |
| pasdly\_cnt | | | | |
|  | | <0,1 – ffffh> | 0が設定された場合、1にする | |
|  | |  |  | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_NONEVNNO 条件未設定のイベント番号を操作した

EX\_PE\_PASOVER パスカウンタ条件設定数オーバー

EX\_SE\_IRGEVNSET 該当するイベント番号が設定されていない

EX\_PE\_INVEVNCOND ハードに設定できないイベント設定を行った

EX\_PE\_SEQOVER 統合イベント設定数オーバー

EX\_SE\_TRACEON トレース実行中

EX\_SE\_TIMERON 時間計測中

EX\_PE\_M\_EVNUSED 指定イベントは現在使用されている

EX\_PE\_SLVEVNNUMUSED 指定イベントは現在違うスレーブ資源で使用されている

EX\_DE\_SELEVTSLAVENOMOUNT 選択されたスレーブ資源が実装されてない、または未サポートである

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [引数説明] | | | | |
|  | | | | |
| \*event1, \*event2, \*event3, \*event4, \*del\_ev | | | | |
|  | <0001h-3fefh> | | シーケンシャルイベントに対応する統合イベント検出器は1個。  1個を超える設定が行われた場合はエラー(EX\_PE\_SEQOVER)を返す。 | |
|  | <4000h-7fdfh> | | 未サポート。  シーケンシャルイベントに対応する統合イベント検出器は1個。  1個を超える設定が行われた場合はエラー(EX\_PE\_SEQOVER)を返す。 | |
|  | \*event1～\*event4、\*del\_evの設定組み合わせは自由に行える。  (例：\*event1の1段のみ、\*event2、\*event3、\*event4の3段、等)  <TEU\_CPU>検出器専用  TEU\_CPU用のシーケンシャルイベント、累積イベントの段数は4段まで設定可能。  また、設定可能なシーケンシャルイベントが1個のみになる。  <TEUスレーブイベント>検出器専用  <TEUスレーブイベント>用のシーケンシャルイベント、累積イベントの段数は3段まで設定可能。  また、設定可能なシーケンシャルイベントがTEUスレーブ用にある<TEU\_TCM>、<TEU\_GRM/TEU\_CRM>、<TEU\_GIO>、<TEU\_GVC/TEU\_KVC/TEU\_AXI/TEU\_DMA>、<TEU\_EMS>、<TEU\_LRM>それぞれに1個ずつ設定が可能である。  設定された段数やシーケンシャルイベントの個数は超えた場合エラー(EX\_PE\_SEQOVER)を返す。  また、<TEU\_CPU>用の単体イベントと<TEUスレーブイベント>の単体イベントを併合することができない。  TEUスレーブイベントの場合でも、異なるスレーブイベント資源(<TEU\_TCM, TEU\_GRM, TEU\_CRM, TEU\_GIO, TEU\_GVC, TEU\_KVC, TEU\_AXI, TEU\_DMA, TEU\_EMS, TEU\_LRM>)の単体イベントを併合することができない。  \*event1、\*event2、\*event3、\*event4、\*del\_evにおいては、違う種類の検出器で設定された単体イベントが併合した場合、エラー(EX\_PE\_INVEVNCOND)を返す。 | | | |
|  |  | | | |
| seq\_cond | | | |  |
|  | bit0 | シーケンシャルイベントを利用 | | 0：無効、1：有効 |
|  | bit1 | 累積ANDイベントを利用 | | 0：無効、1：有効 |
|  | ※シーケンシャルイベントと累積ANDイベントは排他使用のため、bit0とbit1と両方が1の場合、エラー(EX\_PE\_DATA)を返す。なお、bit0とbit1と両方が0の場合、エラーにしないがシーケンシャルイベントを利用することになる。 | | | |
|  |  |  | |  |
|  | bit2 | パスカウントを利用 | | 0：無効、1：有効 |
|  | bit3 | ディレイカウントを利用 | | 0：無効、1：有効 |
|  | ※パスカウント用途とディレイカウント用途は排他使用のため、bit2とbit3と両方が1の場合、エラー(EX\_PE\_DATA)を返す。なお、bit2とbit3と両方が0の場合、エラーにしないがパスカウントを利用することになる。 | | | |
|  |  |  | |  |
|  | Bit8-15 | 未使用 | | 0固定 |
|  |  | | | |
| pasdly\_cnt | | | | |
|  | パスカウントまたはディレイカウント値を指定する。  Seq\_condによって、pasdly\_cntをパスカウント値またはディレイカウント値として用いる。  パスカウント、ディレイカウントの詳細は「動作説明」を参照すること。 | | | |
|  |  | | | |

|  |
| --- |
| [動作説明] |
| 単体イベントは既にex\_rh\_evncondによって設定済みのものが対象となる。設定されていないCPU単体イベントを設定した場合はエラー(EX\_SE\_NONEVNNO)を返し、処理は行わない。  \*event1～\*event4、\*del\_evそれぞれに対し、CPU単体イベントは複数設定できる。複数設定されているCPU単体イベントはORの関係であり、例えばevent1に複数の単体イベントを設定した場合、設定したどれかの単体イベントが成立するとevent1が成立することになる。  シーケンシャルイベントとは、\*event1 -> \*event2 -> \*event3 -> \*event4の順に単体イベントの発生で状態が変化して最終段のイベントが発生で条件が成立する。最終段のイベントが発生する条件が成立する前に\*del\_evで指定したイベントが発生すると、すべての単体イベント(\*event1～\*event4)が再び待ちの状態に初期化される。また、最終段のイベントが発生する条件が成立するとパスカウントの値がデクリメントされ、パスカウントの値が0になるとシーケンシャルイベントが発生する。TEUにはパスカウントのリロード機能があり、V850E2では有効固定にしていてRH850も同様とする。リロード機能を有効にするとシーケンシャルイベント条件成立後にパスカウントの値がリロードされ、シーケンシャルイベントの検出を再実行する。リロード機能を無効にした場合は、シーケンシャルイベントは1回しか検出されない。  累積ANDイベントはシーケンシャルイベントと違って、\*event1、\*event2、\*event3、\*event4のイベントの成立順が関係なく、\*event1、\*event2、\*event3、\*event4の1個ずつのイベントが独自で成立し、すべてのイベント(\*event1、\*event2、\*event3、\*event4)が成立したら累積ANDイベントが発生する。なお、ディレイカウントに指定するCPUクロックはコアごとに異なるが、ディレイカウントを使用したときは、指定カウント値をシステムクロック(V850E2でのHEAPCLK)数として設定し、シーケンシャルイベントもしくは累積ANDイベントの条件成立後、設定したカウント値が経過するとシーケンシャルイベントもしくは累積イベントが発生する。  ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。  (次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に設定可能とする) |
|  |
| 設定[EX\_SET]：統合イベント番号で示したイベント条件を設定する |
| ユーザプログラム実行中でも設定の変更は可能。ただし、トレース関係のイベントに使われているものは、トレース実行中の変更は不可能。統合イベントの使用状況だけでなく、統合イベントに設定されているCPU単体イベントの使用状況も関係する。タイマの開始/終了イベント、および、パフォーマンス計測機能の開始/終了イベントに設定済みの統合イベントをユーザプログラム実行中に変更/削除すると、エラー(EX\_SE\_USRPGMRUN)を返す。  ＜統合イベント条件に使用できないイベント＞  CPUコアイベント(EX\_RUNB\_STS、EX\_RUNB\_XX\_STS)、\*del\_evのみの設定はイベントが発生しないので意味がない。ただし、EXECではエラーにはしない。 |
|  |
| 参照[EX\_REF]：統合イベント番号で示したイベント条件を参照する |
| 未設定の統合イベント番号を指定するとエラー(EX\_SE\_IRGEVNSET)を返す |
|  |
| 削除[EX\_DEL]：統合イベント番号で示したイベント条件を削除する |
| 未設定の統合イベント番号を指定してもエラーとはしない。アクションに設定されている統合イベント番号の削除を行った場合、アクションに設定されたイベント番号がアクションからも削除されるので注意すること |
| 参照/削除のデータ入力時、イベント条件は未設定(ただし、ポインタ不定は厳禁)でよい。 |
|  |
| E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。 |

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

|  |
| --- |
| [マルチコア時の注意点] |
| seq\_noに設定した統合イベント番号が現在選択しているPE以外で設定済みの統合イベント番号だった場合は、funcの指定に関わらずエラー(EX\_PE\_M\_EVNUSED)を返し処理は行わない。スレーブイベントの場合も同じく、noに設定したシーケンシャルイベント番号のイベントが違うスレーブ資源のイベントとして設定されている場合、エラー(EX\_PE\_SLVEVNNUMUSED)を返し処理は行わない。 |

|  |
| --- |
| [初期停止・スタンバイモードデバッグ環境が有効時の注意事項] |
| ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、設定/削除した場合は、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して要求を受け付けない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。 |

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| TEUスレーブ用のシーケンシャルイベント条件を設定する際は、必ずex\_socunitinfo2(EX\_UNIT\_SLAVEIDSEL)より、対象となるスレーブ資源を選択してください。  なお、Phase2以降、TEU\_CPUもしくはコアイベントのシーケンシャルイベント条件を設定するときは、ex\_socunitinfo2(EX\_UNIT\_SLAVEIDSEL)より、スレーブ資源を「0」に戻す必要がある。  スレーブ資源にTEU\_CRMを選択した際は、必ずex\_socunitinfo(EX\_UNIT\_CLUSTERSEL)より、対象となるクラスタ番号を選択してください。 |
|  |

表 3‑13統合イベントの設定例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | \*event1 | \*event2 | \*event3 | \*event4 | \*del\_ev | イベント成立条件 |  |
|  | パスカウントを使うとき | OR/単体 | ------- | ------- | ------- | ------- | \*event1の単体イベント或いはOR条件である単体イベント成立後パスカウントが0になったらイベントが成立する。 |  |
|  | ディレイカウントを使うとき | OR/単体 | ------- | ------- | ------- | ------- | \*event1の単体イベント或いはOR条件である単体イベント成立後設定されたクロック数が経過するとイベントが成立する。 |  |
|  | シーケンシャルイベント条件として使うとき | OR/単体 | OR/単体 | OR/単体 | OR/単体 | OR/単体 | \*event1～\*event4の単体イベントが\*event1->\*event2->\*event3->\*event4の順にイベントが発生するとシーケンシャルイベントが成立する。\*del\_evが発生すると、これまで成立していたシーケンシャル条件はリセットされ、\*event1成立の待ち状態に戻る。 |  |
|  | 累積ANDイベント条件として使うとき | OR/単体 | OR/単体 | OR/単体 | OR/単体 | OR/単体 | \*event1～\*event4の単体イベントが、順番に関係せず、すべてのイベントが発生すると累積ANDイベントが成立する。\*del\_evが発生すると、すべての\*event1～\*event4の成立条件はリセットされ、\*event1～\*event4成立の待ち状態に戻る。 |  |
|  | | | | | | | | |
| 単体：既にex\_rh\_evncondによって設定済みの1個の単体イベント。  OR　：\*event1～\*event4、\*del\_evそれぞれに対し、OR条件として複数単体イベント。  OR条件として複数単体イベントを設定することができ、そのときは設定したどれかが成立するとevent1-4、del\_evが成立することになる。 | | | | | | | | |

### 実行

ユーザプログラムの実行およびブレークの設定/参照/削除の各関数説明を行う。

ユーザプログラムを実行する。

ex\_run

リセット付きでユーザプログラムを実行する。

ex\_resetrun

ユーザプログラムをステップ実行する。

ex\_step

ユーザプログラムを強制ブレークする。

ex\_stop

イベントブレークを設定/参照/削除する。

ex\_hwbrkevent

ソフトウェアブレーク・トリガ設定/参照する。

ex\_swbrkcond

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_run | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **／** | **TRC2** | **／** |

[Function]

ユーザプログラムを実行する

[Format]

int ex\_run( UCHAR mode, ULONG saddr, ULONG eaddr )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| UCHAR | mode | <I > | 実行状態の制御 | ○ |
| ULONG | saddr | <I > | 実行開始アドレス | ○ |
| ULONG | eaddr | <I > | テンポラリブレークアドレス | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mode | | | | | | | | |
|  | 実行条件を設定する | | | | | | |  |
|  | EX\_START | | カレントPCからのスタート | | | | | ① |
|  | EX\_SAD\_ST | | 指定アドレスからのスタート | | | | | ② |
|  | EX\_EBRK\_ST | | カレントPCからのイベントブレーク付きスタート | | | | | ③ |
|  | EX\_SAD\_EB\_ST | | 指定アドレスからのイベントブレーク付きスタート | | | | | ④ |
|  | EX\_SBRK\_ST | | カレントPCからのソフトブレーク付きスタート | | | | | ⑤ |
|  | EX\_SAD\_SB\_ST | | 指定アドレスからのソフトブレーク付きスタート | | | | | ⑥ |
|  | EX\_BRK\_ST | | カレントPCからのイベントブレークおよびソフトブレーク付きスタート | | | | | ⑦ |
|  | EX\_SAD\_BRK\_ST | | 指定アドレスからのイベントブレークおよびソフトブレーク付きスタート | | | | | ⑧ |
|  | EX\_TMPBRK\_ST | | カレントPCからのテンポラリブレーク付きスタート | | | | | ⑨ |
|  | EX\_SAD\_TB\_ST | | 指定アドレスからのテンポラリブレーク付きスタート | | | | | ⑩ |
|  |  | |  | | | | | |
|  | mode指定による実行の状態 | | | | | | | |
|  |  | | フリーRUN | ブレーク付きRUN | | | テンポラリブレーク  eaddr | |
|  | イベント  ブレーク | ソフト  ブレーク | 両方 |
|  | 実行開始  アドレス | PC | ① | ③ | ⑤ | ⑦ | ⑨ | |
|  | saddr | ② | ④ | ⑥ | ⑧ | ⑩ | |
|  |  | |  | | | | | |
| saddr | | | | | | | | |
|  | <address value> | | 実行開始アドレス | | | | | |
|  |  | |  | | | | | |
| eaddr | | | | | | | | |
|  | <address value> | | テンポラリブレークアドレス | | | | | |
|  |  | |  | | | | | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_UE\_RESET リセット継続中で実行できない

EX\_SE\_ACC\_RUN\_PEMODE フラッシュ書き換えモード中(P/Eモード中)のため、フラッシュ資源の読み出しおよび、書き込みはできない

EX\_SE\_ACC\_FCUFSEL\_FCU FCUファーム領域選択がFCUファームウェア資源を指定しているため、読み出しおよび、書き込みはできない

EX\_DE\_CANTACC\_CFLSETTING アクセスできないコードフラッシュ配置に設定されている

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

[Description]

現在のプログラムカウンタ、または、指定したアドレスからユーザプログラムの実行を行う。

また、テンポラリブレークのアドレスも設定できる。テンポラリブレークはex\_hwbrkevent()やex\_swbrkcond()を無効にし、1回のみ有効なブレーク設定である。

既にユーザプログラム実行中の場合には、エラー(EX\_SE\_USRPGMRUN)を返し処理は行わない。

実行開始アドレスがソフトブレークポイントか実行前PCブレークポイントの場合、そのアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。テンポラリブレークも実行前PCブレークポイントと同じ動きになる。テンポラリブレークのsaddr=eaddrの場合、saddr指定のアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。

何らかの問題でリセット信号が入りっぱなしの時は実行できないので、エラー(EX\_UE\_RESET)を返す。CPUリセットをマスクすればとりあえず動かすことは可能。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

操作対象のPEが停止(初期停止・Cyclic Disable)中の場合、エラー応答(EX\_SE\_CPUSTOP\_NOSPRT)して要求を受け付けない。

操作対象のPEが停止(初期停止・Cyclic Disable)中でない場合、要求を受け付けて実行を開始する。戻り値は正常終了を返す。

ただし、実行開始時の状態によっては以下の制限が発生する。

・「同期デバッグモード」かつ「CPUコア(実行前)イベントが設定されている」かつ「停止(初期停止・Cyclic Disable)中のPEが存在する」の状態で、イベントブレーク付き実行を指定した場合、停止(初期停止・Cyclic Disable)中のPEに対するCPUコア(実行前)イベントは設定できない。

そのため、起動しているPEに対してのみCPUコア(実行前)イベントを設定して実行開始する。

・「同期デバッグモード」かつ「CPUコア(実行前)イベントが設定されている」かつ「停止(初期停止・Cyclic Disable)中のPEが存在する」の状態で、フリー実行を指定した場合、停止(初期停止・Cyclic Disable)中のPEに設定済みのCPUコア(実行前)イベントの削除ができない。

そのため、起動しているPEに対してのみCPUコア(実行前)イベントを削除して実行開始する。

・Cyclicモード中でソフトウェアブレーク付き実行を指定した場合、Flash資源に対するソフトウェアブレークの設定ができない。

そのため、ソフトウェアブレークの設定を行わず実行開始する。

・Cyclicモード中でフリー実行を指定した場合、Flash資源に対するソフトウェアブレークの削除ができない。

そのため、ソフトウェアブレークの削除を行わず実行開始する。

上記、CPUコア(実行前)イベント、ソフトウェアブレークの設定・削除が行えないことにより、意図したイベント・ソフトウェアブレークが発生しない、その逆で意図しないイベント・ソフトウェアブレークが発生する可能性がある。なお、意図しないイベント・ソフトウェアブレークが発生した場合のブレーク要因は、従来通りのイベントブレークもしくはソフトウェアブレークとなる。

|  |  |
| --- | --- |
| [マルチコア時の注意点] | |
| ① PEの選択に関わらずPE同時実行を行う。また、CPUコアの一斉ブレークを有効にして実行する。ただし、各PEが実行されるタイミングを同じにすることはできない | |
|  |  |
| ② 開始アドレス指定、テンポラリブレークは選択PEのみ有効となる。 | |
|  |  |
| ③ PFC1Aの同期ブレーク機能について：  同期ブレークを利用するためにリレーブレーク機能があるが、PFC1Aはリレーブレーク機能にて一部不具合がある。リレーブレーク機能を使ってPE3からPE1へ同期ブレークをさせるには問題はないが、PE1からPE3へ同期ブレークさせるにはリレーブレーク機能は使用できない。  PE1からPE3へ同期ブレークをさせるための対策としてTEUのクロストリガ機能を使う。ただし、CPUのハードウェアブレークを利用したときは、クロストリガ機能は利用できない。またTEUのイベントブレークのときもブレークポイントの位置によって、PE3へブレークさせるにはクロストリガ機能が効かない場合がある。クロストリガ機能が利用できないことに備えて、PE1からPE3へ同期ブレークさせるために強制ブレーク機能を利用する。同期ブレークのためPE3へ強制ブレークをかけるのは、PE1がブレークしてからの処理になるため、PE3の方が遅れてブレークする。  PFC1Aでの同期ブレーク機能をまとめると以下のようになる。  コアブレークの場合  PE1 -> PE3：強制ブレーク「クロストリガ機能使用不可」 (要因はリレーブレーク)  PE3 -> PE1：リレーブレーク機能 (要因はリレーブレーク)  TEUブレークの場合  PE1 -> PE3：クロストリガ機能もしくは強制ブレーク (要因はリレーブレーク)  PE3 -> PE1：リレーブレーク機能 (要因はリレーブレーク) | |

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| Phase2以降でソフトウェアブレーク機能(ex\_swbrkcond I/F)をサポートするため、ex\_runのmode引数においてはEX\_SBRK\_STおよびEX\_SAD\_SB\_STを指定した場合エラーは返さない。 |
| 実行開始アドレスがソフトブレークポイントか実行前PCブレークポイントの場合、そのアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。テンポラリブレークも実行前PCブレークポイントと同じ動きになる。テンポラリブレークのsaddr=eaddrの場合、saddr指定のアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。  1ステップ実行を行うときは、一旦すべてのブレークイベント(TEUおよびコア)を無効にし、実行PCにソフトウェアブレークポイントがある場合はソフトウェアブレーク命令をユーザコードに置き換えをして、1ステップ実行が完了してからブレークイベントやソフトウェアブレーク命令を元に戻す。このため、1ステップ実行中は、設定しているブレークイベントは検出されない。 |
| G4世代デバイスでは、セキュリティコアを有効にした動作モード時、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置に切り替わっている可能性がある。  セキュリティコアを有効にした動作モードでかつ同期デバッグモードの場合、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置であればエラー(EX\_DE\_CANTACC\_CFLSETTING)を返す。 |

|  |
| --- |
| [FCUに関連する注意事項] |
| メモリアクセスするフラッシュ資源がP/Eモード中は、メモリアクセスを行わずEX\_SE\_ACC\_RUN\_PEMODEを返送する。  FCUファーム領域選択レジスタがFCUファームウェア資源を指定し、コードフラッシュを対象としてメモリアクセスを行った場合は、メモリアクセスを行わずEX\_SE\_ACC\_FCUFSEL\_FCUを返送する。 |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_resetrun | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

リセット付きでユーザプログラムを実行する

[Format]

int ex\_resetrun(UCHAR mode, ULONG eaddr)

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| UCHAR | mode | <I > | 実行状態の制御 | ○ |
| ULONG | eaddr | <I > | テンポラリブレークアドレス | ○ |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| mode | | | |
|  | 実行条件を設定する | |  |
|  | EX\_START | カレントPCからのスタート |  |
|  | EX\_EBRK\_ST | カレントPCからのイベントブレーク付きスタート |  |
|  | EX\_SBRK\_ST | カレントPCからのソフトブレーク付きスタート |  |
|  | EX\_BRK\_ST | カレントPCからのイベントブレークおよびソフトブレーク付きスタート |  |
|  | EX\_TMPBRK\_ST | カレントPCからのテンポラリブレーク付きスタート |  |

|  |  |  |
| --- | --- | --- |
| eaddr | | |
|  | <address value> | テンポラリブレークアドレス |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_SE\_ANOTHERPERUN 別のコアがRUN中のため実行できない

EX\_SE\_ASPENABLE ASPが有効

EX\_SE\_SFTTRACEENB 外部ソフトトレース単線出力が有効

EX\_SE\_NOSPRT\_DURING\_CANBOOT CANブートデバッグ中のため本関数は使用できない

EX\_DE\_CANTACC\_CFLSETTING アクセスできないコードフラッシュ配置に設定されている

[Description]

リセット状態でユーザプログラムを実行開始し、RUN状態に遷移後にリセットを解除し、ユーザプログラムをリセット動作から開始する。

全コアを対象にリセットおよび、プログラム実行状態へ遷移させる。全コアを対象にするため、別のコアがユーザプログラム実行中の場合はエラー(EX\_SE\_ANOTHERPERUN)を返す。

E2拡張機能が有効な状態で本コマンドを実行すると、エラー(EX\_SE\_CANTSET\_ASPON)が返る。

E2拡張機能が無効な状態かつ外部ソフトトレース単線出力機能が有効な状態で本コマンドを実行すると、エラー(EX\_SE\_SFTTRACEENB)が返る。

CANブートデバッグ時に本コマンドを実行すると、エラー(EX\_SE\_NOSPRT\_DURING\_CANBOOT)が返る。

G4世代デバイスでは、セキュリティコアを有効にした動作モード時、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置に切り替わっている可能性がある。

セキュリティコアを有効にした動作モードでかつ同期デバッグモードの場合、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置であれば、エラー(EX\_DE\_CANTACC\_CFLSETTING)を返す。

[非同期デバッグモード時の注意点]

非同期デバッグモードではソフトウェアブレークは使用できないため、modeにEX\_SBRK\_STもしくはEX\_BRK\_EXを設定しても、ソフトウェアブレークしない。

[デバッガ開発者向け注意事項]

初期停止・スタンバイモードデバッグ環境が有効時、EXEC内部で一旦全てのPEを起動させ、必要なCPUコア(実行前)イベントブレーク・ソフトウェアブレークの設定を行ってから、リセットによる初期停止状態へ戻して実行開始する。

このため、初期停止・スタンバイモードデバッグ環境が有効時であっても、従来のデバッグ環境のモードと同様に、ex\_resetrunとex\_reset＋ex\_runは等価ではない。

そのため、ユーザへリセット後RUNの機能を提供する場合は、デバッガ側でex\_resetとex\_runを順に呼ぶことで実現するのではなく、ex\_resetrunを使用することを推奨する。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_step | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **／** | **TRC2** | **／** |

[Function]

ユーザプログラムをステップ実行する

[Format]

int ex\_step( UCHAR mode, USHORT count, ULONG saddr, ULONG rsaddr, ULONG readdr )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| UCHAR | mode | <I > | 実行状態の制御 | ○ |
| USHORT | count | <I > | ステップ実行回数 | ○ |
| ULONG | saddr | <I > | 実行開始アドレス | ○ |
| ULONG | rsaddr | <I > | 範囲指定下限値アドレス | ○ |
| ULONG | readdr | <I > | 範囲指定上限値アドレス | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| mode | | | | | | |
|  | 実行条件を設定する | | | | |  |
|  | EX\_CNT\_STP | | カレントPCからのステップ | | | ① |
|  | EX\_SAD\_CNT\_STP | | 指定アドレスからのステップ | | | ② |
|  | EX\_RAN\_STP | | カレントPCから指定範囲内をステップ | | | ③ |
|  | EX\_SAD\_RAN\_STP | | 指定アドレスから指定範囲内をステップ | | | ④ |
|  | EX\_UNTIL\_STP | | カレントPCから指定範囲内になるまでステップ | | | ⑤ |
|  | EX\_SAD\_UNTIL\_STP | | 指定アドレスから指定範囲内になるまでステップ | | | ⑥ |
|  |  | |  | | | |
|  | mode指定によるステップ実行の状態 | | | | | |
|  |  | | プロシジャーステップ | | | |
|  |  |  | なし | あり | あり+イベント有効 | |
|  | アドレス  範囲指定 | 開始  アドレス |
|  | なし | PC | ① | ---- | ---- | |
|  | saddr | ② | ---- | ---- | |
|  | rsaddr、reaadr  (範囲内) | PC | ③ | ---- | ---- | |
|  | saddr | ④ | ---- | ---- | |
|  | rsaddr、reaadr  (UNTIL) | PC | ⑤ | ---- | ---- | |
|  | saddr | ⑥ | ---- | ---- | |
|  |  | |  | | | |
| saddr | | | | | | |
|  | <address value> | | 実行開始アドレス | | | |
|  |  | |  | | | |
| rsaddr | | | | | | |
|  | <address value> | | 範囲指定下限値アドレス | | | |
|  |  | |  | | | |
| readdr | | | | | | |
|  | <address value> | | 範囲指定上限値アドレス | | | |
|  |  | |  | | | |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

~~EX\_UE\_BROMAREA 裏ROMエリアを指定した~~

EX\_UE\_ENFBRK 強制ブレークできない

EX\_SE\_ACC\_RUN\_PEMODE フラッシュ書き換えモード中(P/Eモード中)のため、フラッシュ資源の読み出し

および、書き込みはできない

EX\_SE\_ACC\_FCUFSEL\_FCU FCUファーム領域選択がFCUファームウェア資源を指定しているため、読み出しおよび、書き込みはできない

EX\_DE\_CANTACC\_CFLSETTING アクセスできないコードフラッシュ配置に設定されている

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

[Description]

ステップ実行を行う。ステップ終了条件としてcount数およびアドレス範囲が設定可能である。プロシジャーステップはサポートしていない。

count数を指定した場合、ユーザプログラムをcount数回ステップ実行する。countの値が0のときは、エラー(EX\_PE\_DATA)を返す。範囲指定ステップの場合でも、必ずカウント条件(count)を指定する。

EX\_RAN\_STP、EX\_SAD\_RAN\_STPのアドレス範囲を指定した場合、範囲以外を実行する前にブレークする。なお、スタートアドレスがアドレス範囲以外の場合は何も実行しない。

EX\_UNTIL\_STP、EX\_SAD\_UNTIL\_STPのアドレス範囲までのステップ実行を指定した場合、アドレス範囲内に到達した場合にブレークする。ポイント指定の場合は、rsaddr、readdrを同じにすること。なお、スタートアドレスがアドレス範囲内の場合でも必ず1ステップ行い、その後のアドレスからブレークの判定を行う。

rsaddr＞readdrの場合はエラー(EX\_PE\_DATA)を返す。

既にユーザプログラム実行中の場合には、エラー(EX\_SE\_USRPGMRUN)を返し処理は行わない。

G3コアの場合、CPUコアのステップ機能を使うためpswのbit11を操作する。割り込み処理が起きないようにpswのNPフラグも操作する。

スタンバイモードに入った場合は、スタンバイをキャンセルしてステップ処理を続ける。

マルチコアで同期デバッグモードの場合、全コアを同時に1ステップずつ実行する。マルチコアで同期デバッグモードの場合、ex\_stepのすべての引数(mode、count、saddr、raddr、reddr)はカレントコアだけが対象となる。カレントコア以外のコアは、カレントPCから1ステップずつ実行することになる。

※Phase1ではマルチコアの場合でもカレントコアのみがステップ実行し、その他のコアはブレーク状態のままであった。

G4世代デバイスでは、セキュリティコアを有効にした動作モード時、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置に切り替わっている可能性がある。

セキュリティコアを有効にした動作モードでかつ同期デバッグモードの場合、ユーザのデバイス設定によってはコードフラッシュ領域の配置がアクセスできない配置であれば、エラー(EX\_DE\_CANTACC\_CFLSETTING)を返す。

G4世代デバイスで仮想化支援機能をサポートしている場合、ex\_socunitinfo2のEX\_UNIT\_HEGESELでデバッグ対象CPU動作モードの指定やステップ機能オプションが指定できる。ステップ機能オプションを有効に設定すると、デバッグ対象のCPU動作モード内でステップ実行が可能となる。ステップ実行によってデバッグ対象外のCPU動作モードに遷移した場合は、デバッグ対象外のCPU動作モードではブレークせずに、デバッグ対象のCPU動作モードになるまで実行を継続した後にブレークする。本機能の対象はカレントコアのみである。同期デバッグモードにおける非カレントコアはカレントコアと同時に実行を開始し、カレントコアのブレークに伴うリレーブレーク要求を受けてブレークする。カレントコアが暴走などにより、デバッグ対象のCPU動作モードに遷移しない場合を考慮して、3秒待ってブレークしない場合はタイムアウトとして扱い強制ブレークする。タイムアウトした場合、ex\_getstatusのexestat – es\_vm\_timeout(bit14)が1となる。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

操作対象のPEが停止(初期停止・Cyclic Disable)中の場合、要求を受け付けずエラー応答(EX\_SE\_CPUSTOP\_NOSPRT)する。

操作対象のPEが停止(初期停止・Cyclic Disable)中でない場合、要求を受け付けてステップ実行を開始するが、同期デバッグモード時は、起動中のPEのみステップ実行を行い、初期停止状態またはCyclic Disable状態のPE対してはステップ実行を行わない。ただし、起動中のPEの連続ステップ実行により初期停止状態またはCyclic Disable状態が解除されるとその時点からステップ実行を開始する。

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| ex\_stepコマンド発行後、指定されたステップ条件がすべて完了するまでは何も返ってこない。  なお、範囲指定ステップの場合、必ずカウント条件(count)を指定し、様子をみながらex\_step()を呼ぶような使い方をしてもらう必要がある。  指定されたステップ条件がすべて完了した場合はEX\_NOERRORを返す。ただし、ステップ実行で失敗した場合は、強制ブレークをかけてエラー扱いにするため(EX\_FE\_MONTIMEOUT)を返す。ステップ実行で失敗し、強制ブレークをかけてもブレークできない場合はエラー(EX\_UE\_ENFBRK)を返す。マルチコアで同期デバッグモードの場合は全コア同時ステップするため、全コアがブレークに成功した場合のみEX\_NOERRORを返す。 |
| ステップ実行を行うときは、一旦すべてのブレークイベント(TEUおよびコア)を削除し、ステップ実行が完了してから改めて、削除したブレークイベントの条件を設定する。そのため、ステップ実行中は設定しているブレークイベントが検出できない。 |
| Phase2.0までは、シングルステップ実行中下記の例外が発生したときは2命令実行してブレークしていたが、Phase2.1以降は1命令実行してブレークする。また、下記の命令からステップ実行した場合でも2命令実行してブレークしていたのは、Phase2.1以降は1命令実行してブレークすることになる。  2命令実行する例外一覧 2命令実行する命令一覧  SYSERR(システム・エラー例外) TRAP命令  RIE(予約命令例外) SYSCALL命令  UCPOP(コプロセッサ例外) HVTRAP命令  MIP(メモリ保護例外(実行権)) CALLT命令  MDP(メモリ保護例外(アクセス権)) FETRAP命令  ITLBE(TLB例外(実行権)) HVCALL命令  DTLBE(TLB例外(アクセス権)) RIE命令  PIE(特権命令例外) MAE(ミスアライン例外)  FENMI例外 FPP(FPU例外(プレサイス)) |

|  |
| --- |
| [マルチコア時の注意点] |
| 同期デバッグモード時は全コアが同時にステップ実行を行う。  非同期デバッグモード時は選択コアのみがステップ実行を行う。 |
|  |

|  |
| --- |
| [FCUに関連する注意事項] |
| メモリアクセスするフラッシュ資源がP/Eモード中は、メモリアクセスを行わずEX\_SE\_ACC\_RUN\_PEMODEを返送する。  FCUファーム領域選択レジスタがFCUファームウェア資源を指定し、コードフラッシュを対象としてメモリアクセスを行った場合は、メモリアクセスを行わずEX\_SE\_ACC\_FCUFSEL\_FCUを返送する。 |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_stop | **SYNC** | **EMU** | **○** | **BRK** | **×** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

ユーザプログラムを強制ブレークする

[Format]

int ex\_stop( void )

[Arguments]

なし

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_SE\_USRPGMSTOP ユーザプログラムブレーク中

EX\_UE\_ENFBRK 強制ブレークできない

EX\_SE\_FETCHSTOP\_BEFORE\_BRKFAILED 初期停止中のため、ブレーク要求を出さずに処理を中断した

EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED　 初期停止中のため同期ブレークできませんでした

EX\_SE\_CYCLIC\_BEFORE\_BRKFAILED Cyclicモード中のため、ブレーク要求を出さずに処理を中断した

EX\_SE\_CYCLIC\_AFTER\_BRKFAILED Cyclicモード中のため同期ブレークできませんでした。

EX\_SE\_STANDBY\_NOSPRT　　 該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

実行中のユーザプログラムを強制的にブレークさせる。

既にユーザプログラムブレーク中の場合には、エラー(EX\_SE\_USRPGMSTOP)を返し、処理は行わない。

本関数を呼び出しても強制ブレークすることができない場合にはエラー(EX\_UE\_ENFBRK)を返す。

[初期停止・スタンバイモードデバッグ環境が無効時の注意事項]

同期デバッグモードにおいて、マルチコアでいずれかのコアが初期停止中の場合はエラー(EX\_SE\_FETCHSTOP\_BEFORE\_BRKFAILED)を返す。非同期デバッグモードにおいては、初期停止中のコアに対して呼び出した場合に限ってエラー(EX\_SE\_FETCHSTOP\_BEFORE\_BRKFAILED)を返す。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)通常接続後のユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

(2)ホットプラグイン接続後(ホットプラグインRUN中)に、停止(初期停止・Cyclic Disable)中のコアが存在した場合、強制ブレーク要求に対して以下のエラーを返す。

・初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

・初期停止中の場合は、エラー(EX\_SE\_FETCHSTOP\_BEFORE\_BRKFAILED)を返す。

・CyclicRunモード中の場合は、エラー(EX\_SE\_CYCLIC\_BEFORE\_BRKFAILED)を返す。

・デバッガからの強制ブレーク要求を受け付けた後に、停止(初期停止・Cyclic Disable)中を検出した場合は、要求受け付け後のエラー(EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED、EX\_SE\_CYCLIC\_AFTER\_BRKFAILED)を返す。

[マルチコア時の注意点]

同期デバッグモード時はPEの選択に関わらずPE同時強制ブレークを行う。ただし、各PEがブレークするタイミングを同じにすることはできない。

非同期デバッグモード時は選択PEのみ強制ブレークを行う。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_hwbrkevent | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **○** |

[Function]

イベントブレーク条件を設定/参照/削除する

[Format]

int ex\_hwbrkevent( USHORT func, EXHEBRK \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| EXHEBRK | \*cond | <IO> | イベント条件 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | num | <IO> | イベント設定数 | ○ |
|  | USHORT | \*event | <IO> | 設定数個のイベント番号情報 | ○ |
| }EXHEBRK | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | イベントブレーク条件を設定する |
|  | EX\_REF | イベントブレーク条件を参照する |
|  | EX\_DEL | イベントブレーク条件を削除する |
|  | EX\_BRK\_REF | ブレークが成立したイベントを参照する |
|  |  |  |
| num | | |
|  | <0,1 – evn\_num> | 0はEX\_SET、EX\_DELでは設定不可。 |
|  |  |  |
| \*event | | |
|  | <event number> | 上位側のソフトで用意されているイベント番号用のバッファのポインタ |
|  |  |  |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_NONEVNNO 条件未設定のイベント番号を操作した

EX\_PE\_HWBRKNUMOVER 指定イベントの個数が設定可能なイベント個数を超えている

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_CPUSTOP\_NOSPRT CPUが停止中であるためコマンド実行ができない

EX\_SE\_STANDBY\_NOSPRT　　　該当するスタンバイモード中はデバッグ操作を行なえないためコマンド実行ができません。デバッグを継続する場合スタンバイモードが解除された状態での操作もしくは強制リセットを行ってください。

[Description]

|  |  |
| --- | --- |
| イベントをブレーク条件に割り当てる。 | |
| 設定：イベント設定数分のイベント番号をブレーク条件として設定する。 | |
|  | 既に設定されているイベントに追加される。 |
|  | |
| 参照：ブレーク条件として設定されているイベント数とイベント番号を返す。 | |
|  |  |
| 削除：イベント設定数分のイベント番号をブレーク条件から削除する。 | |
|  |  |
| 成立イベント参照：ブレークイベントを発生させたイベント数とイベント番号を返す。 | |
|  |  |
| 設定/削除にてnumに0が指定された場合はエラー(EX\_PE\_DATA)を返し、処理は行わない。  設定を行う場合、既にex\_rh\_evncondとex\_rh\_evnintgcondによって設定済みのイベントのみ対象となり､設定されていないイベントの操作はエラー(EX\_SE\_NONEVNNO)を返し、処理は行わない。  削除を行う場合、設定されていないイベントがあってもエラーにしない。  参照を行う場合、設定してある数に見合う充分大きなバッファのポインタをeventに設定しておくこと。  ユーザプログラム実行中にCPUコアイベントを設定/削除した場合は、強制ブレークで止めて設定し再実行させる。(ブレーク期間はホストPCの時間間隔)  ブレークをさせてはいけない設定をデバッグしているユーザのために、ブレークさせないよう上位側ソフトでCPUコアイベントを設定/削除が発生しないように管理をすること。  ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。  (次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に設定可能とする)  E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。 | |

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

・制御対象PEが停止(初期停止・Cyclic Disable)中の状態で、CPUコア(実行前)イベントを設定/削除した場合は、エラー(EX\_SE\_CPUSTOP\_NOSPRT)を返して設定を受け付けない。

・ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、設定/削除した場合は、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して要求を受け付けない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

|  |
| --- |
| [デバッガ開発者向け注意事項] |
| ex\_getenvでは個数(evn\_num)を返さない仕様とする。上位モジュールでの管理は、設定できなくなるまで設定することで実現することとする。 |
| ex\_runで実行開始アドレスが実行前PCブレークポイントの場合、そのアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。  1ステップ実行を行うときは、一旦すべてのブレークイベント(TEUおよびコア)を削除し、1ステップ実行が完了してから改めて、削除したブレークイベントの条件を設定する。このため、1ステップ実行中は、設定しているブレークイベントが検出できない。 |
| TEUスレーブイベントへブレーク条件を設定する際は、必ずex\_socunitinfo2(EX\_UNIT\_SLAVEIDSEL)より、対象のなるスレーブ資源を選択してください。  なお、Phase2以降、TEU\_CPUもしくはコアイベントへブレーク条件を設定するときは、ex\_socunitinfo2(EX\_UNIT\_SLAVEIDSEL)より、スレーブ資源を「0」に戻す必要がある。 |
| G3MHコアの場合、特定の条件で成立していないイベントが成立となる不具合がある。  そのため、G3MHコアを使用時でfuncにEX\_BRK\_REFを指定した場合、ブレーク要因となったイベント番号  が正しくない場合がある。 |

|  |
| --- |
| [マルチコア時の注意点] |
| 1. CPUコアイベントおよびTEU\_CPUイベントは選択PEのリソースに対して設定される。このため、選択したPEで設定したイベント番号を設定する必要がある。 |
| ② funcにEX\_BRK\_REFを指定した場合は、PEの選択に関係なくブレーク要因となったイベント番号が返る。ブレーク要因となったイベントが選択PE以外のイベント番号である場合があるので、イベント番号の管理をPEごとに行う必要がある。異なるPEのイベント番号だった場合、ex\_rh\_evncondで参照してもエラーが返るので注意が必要。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_swbrkcond | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

ソフトウェアブレーク条件を設定/参照/削除する

[Format]

int ex\_swbrkcond( USHORT func, USHORT \*num, EXSWBRK \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | \*num | <IO> | 設定するソフトウェアブレークポイントの数の指定  設定されているソフトウェアブレークポイントの数 | ○ |
| EXSWBRK | \*cond | <IO> | ソフトウェアブレーク条件 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | sbno | <IO> | ブレークポイント番号 | ○ |
|  | USHORT | sberr | < O> | エラーフラグ | ○ |
|  | ULONG | sbaddr | <IO> | ブレークアドレス | ○ |
|  | UCHAR | sbdata[2] | <--> | ---------- | -- |
|  | UCHAR | dummy[2] | < O>  \*1 | dummy[0]：未使用  dummy[1]：PE番号  \*1 dummy[0]の属性は<-->、dummy[1]の属性は<-O> | ○ |
| }EXSWBRK | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | ソフトウェアブレーク条件を設定する |
|  | EX\_REF | ソフトウェアブレーク条件を先頭から参照する |
|  | EX\_REF\_NEXT | ソフトウェアブレーク条件を指定ポイントから参照する |
|  | EX\_DEL | ソフトウェアブレーク条件を削除する |
|  | EX\_REF\_NUM | ソフトウェアブレーク設定総数の参照 |
|  |  |  |
| num | | |
|  | <1 - 2000> | 設定するソフトウェアブレークポイントの数を指定する。  または、設定されているソフトウェアブレークポイントの数を返す。 |
|  |  |  |
| sbno | | |
|  | <1 - ffffh> | 任意のブレークポイント番号を設定する |
|  |  |  |
| sberr | | |
|  | EX\_SB\_NOERROR | 正常終了 |
|  | EX\_SB\_NUMBERERR | 設定されていないブレークポイントを削除しようとした |
|  | EX\_SB\_MEMERROR | メモリのsbaddr領域に正しくソフトウェアブレークが設定できなかった。ブレークポイントは設定されない |
|  | EX\_SB\_POINTOVER | 最大のブレークポイント数を超えて設定しようとした。ブレークポイントは設定されない |
|  |  |  |
| sbaddr | | |
|  | <address value> | ブレークポイントアドレス |
|  |  |  |
| dummy[0] |  |  |
|  |  | 未使用 |
|  |  |  |
| dummy[1] |  |  |
|  | <PE番号> | ソフトウェアブレーク条件を設定したときに設定されていたPE番号 |

[Return Values]

EX\_NOERROR 正常終了

EX\_FE\_COMM EXECを起動していない

EX\_PE\_DATA 設定値が異常

EX\_SE\_USRPGMRUN ユーザプログラム実行中

EX\_SE\_SLFPRGONSWBRKNOSPRT セルフプログラミングを行う設定が有効であるためソフトウェアブレークの設定ができない

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_NOSPRT\_RAMINITOFF RAM初期化がOFFのためソフトウェアブレークの設定ができない。

EX\_SE\_CANTCALL\_ASPRUNNING 拡張機能を使用したユーザプログラム実行中

EX\_SE\_SFTTRCRUN 外部ソフトトレース単線出力でユーザプログラムを実行中

EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED Cyclicモード中であるためフラッシュ領域へライトできない

[Description]

[引数説明]

|  |  |
| --- | --- |
| func=EX\_SET | \*num個分のブレークポイント番号とブレークポイントアドレスを追加して設定する。設定された情報として、設定されているブレークポイントの総数とブレークポイントごとのエラーフラグを返す。 |
| func=EX\_REF | 設定されているブレークポイントの総数と設定されているブレークポイントの総数分のブレークポイント番号とブレークポイントアドレスを返す。ただし、総数が100個を超える場合は100個目以降の情報は返さない(情報量が多くなりすぎるため)。 |
| func=EX\_REF\_NEXT | 設定されているブレークポイントの総数と指定された番号以降のブレークポイント番号とブレークポイントアドレスを返す。ただし、指定された番号以降の総数が100個を超える場合は100個目以降の情報は返さない(情報量が多くなりすぎるため)。 |
| func=EX\_DEL | \*num個のブレークポイント番号に対応した設定内容を削除する。 |
| func=EX\_REF\_NUM | 設定されているブレークポイントの総数を返す。 |

[動作説明]

ソフトウェアブレーク条件の設定/参照/削除/設定総数参照を行う。

設定時には追加設定を行うが、同一のソフトウェアブレーク番号の場合、上書き設定となる。

設定できるブレークポイントの最大個数は2000個とする。

ソフトウェアブレークにはERAMを使用しない。命令埋め込み方式で実現する。

ソフトウェアブレークはコードフラッシュ領域と一部を除く内蔵RAM領域に対応する。(設定できない内蔵RAM領域の詳細は[デバッガ開発者向け注意事項]を参照)データフラッシュ領域、外部領域などは未対応とする。未対応領域への設定要求があった場合はsberr引数にエラー(EX\_SB\_MEMERROR)を返す。

また、sbaddrに奇数の値を設定した場合もsberr引数にエラー(EX\_SB\_MEMERROR)を返す。

以下ブレーク中とユーザプログラム実行中のときソフトウェアブレークのアドレスを読み出したとき、埋め込んだ命令が見えるか見えないかを示す。

|  |  |
| --- | --- |
| ブレーク中/実行中にメモリを読み出した場合 | |
| ソフトウェアブレーク  方式  状態 | 命令埋め込み方式 |
| ブレーク中 | 見えない |
| 実行中 | 見える |

ユーザプログラム実行中に設定/削除した場合は、何もせずエラー(EX\_SE\_USRPGMRUN)を返す。

ホットプラグインRUN中(接続～強制ブレークするまで)も、EX\_SE\_USRPGMRUNエラーとする。

実行中にソフトウェアブレーク命令を書き換えた場合(リセットによるRAM初期化を含む)、動作の保証はしない。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中に参照(EX\_REF, EX\_REF\_NEXT, EX\_REF\_NUM)で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)に参照(EX\_REF, EX\_REF\_NEXT, EX\_REF\_NUM)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

Cyclicモード中にコードフラッシュ領域に対してソフトウェアブレークを設定/削除した場合は、

エラー応答(EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED)して設定/削除を受け付けない。

また、Cyclicモード中に複数のソフトウェアブレークの設定/削除する指定でコードフラッシュ領域が1つでも入っていた場合は、全ての設定/削除を反映せずエラー応答(EX\_SE\_CYCLIC\_FLASH\_WRITEFAILED)する。

|  |
| --- |
| [マルチコア時の注意点] |
| (1)PEの選択に関わらずPE共通にソフトウェアブレークの設定を行う。ただし、マルチコアデバイスにおいて内蔵RAM self領域のアドレスをソフトウェアブレーク条件とした場合は、現在選択されている物理CPU(PE)の内蔵RAMに対して設定を行う。なお、ソフトウェアブレーク条件を設定したときに設定されていた物理CPU(PE)番号はdummy[1]で参照することができる。  (2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。 |
| [デバッガ開発者向け注意事項] |
| (1)ex\_runで実行開始アドレスがソフトウェアブレークポイントの場合、そのアドレスを1ステップ実行してから通常のユーザプログラム実行処理を行う。1ステップ実行を行うときは、ソフトウェアブレーク命令をユーザコードに置き換えて、1ステップ実行が完了してから、改めてソフトウェアブレーク命令への書き換えを行うため、1ステップ実行中はブレークが発生しない。 |
| (2)ex\_socunitinfo2(EX\_UNIT\_FLPGOPT)より、コードフラッシュのセルフプログラミングを行う指定の場合は、ソフトウェアブレークの設定条件の組み合わせによっては、コードフラッシュ資源へのソフトウェアブレーク設定が不可としてエラー(EX\_SE\_SLFPRGONSWBRKNOSPRT)を返す。設定条件の組み合わせに対する応答は下記の通り。  ・すべてコードフラッシュ領域：エラー(EX\_SE\_SLFPRGONSWBRKNOSPRT)を返す。  ・コードフラッシュ領域と内蔵RAM領域の混在：エラー(EX\_SE\_SLFPRGONSWBRKNOSPRT)を返す。  ・すべて内蔵RAM領域：正常終了(EX\_NOERROR)を返す。 |
| (3)ex\_socunitinfo2のEX\_UNIT\_LPDOPTで「RAM初期化をしない」に設定している場合、エラー(EX\_SE\_NOSPRT\_RAMINITOFF)を返し、処理は行わない。ソフトウェアブレーク命令をフラッシュメモリに書き込む際の内部処理でRAM領域への書き込みにより、意図せぬ初期化や、ECCエラーが発生してしまうため。ソフトウェアブレークを使用する場合は「RAM初期化をする」に設定すること。  (4)D1xのRetention RAM領域への設定は、書き込みが可能な状態の時にだけ可能とする。書き込みが可能でないときに設定要求があった場合は、sberr引数にエラー(EX\_SB\_MEMERROR)を返す。 |

[非同期デバッグモード時の注意点]

非同期デバッグモードで、本関数を呼び出すとエラー(EX\_SE\_ASYNCMODE\_ENB)を返す。非同期デバッグモードではソフトウェアブレークが使用できない。同期デバッグモード中にソフトウェアブレークを設定し、非同期デバッグモードに遷移後プログラム実行してもソフトウェアブレークは発生しない。ただし、この後に再び同期デバッグモードに遷移すると、設定済みのソフトウェアブレークが有効となり、プログラム実行するとソフトウェアブレークによるブレークが発生可能な状態になる。

### Analyzer

The analyzer is divided into three types: “Trace”, “Coverage”, and “Timer / Performance Measurement”, which are controlled by the following functions.

(1) Trace function

ex\_trcsw, ex\_trcrun, ex\_trcmode, ex\_trcclear, ex\_trcevent, ex\_trcsearch, ex\_trcseek,

ex\_trcdelay, ex\_trcread, ex\_m\_trceventsel2, ex\_rh\_trceventsel2, ex\_m\_trcbufcont,

ex\_rh\_trcbufcont, ex\_trcrawmeminfo, ex\_trcrawmemread, ex\_trchwtune, ex\_rh\_trcfilter

(2) External soft trace single line output function

ex\_rh\_sfttrcsw, ex\_rh\_sfttrcmode, ex\_rh\_sfttrcevent, ex\_rh\_sfttrcfilter, ex\_rh\_sfttrctune,

ex\_rh\_sfttrcrun, ex\_rh\_sfttrcrawmemmode, ex\_rh\_sfttrcrawmeminfo, ex\_rh\_sfttrcrawmemread

(3) RRM / カバレッジ関数

ex\_rrmmapping, ex\_rrmsw, ex\_rrmread, ex\_rrmclear

ex\_covmapping, ex\_covsw, ex\_covrun, ex\_covclear, ex\_covread, ex\_covwrite, ex\_covsearch

(4) タイマ関数 / パフォーマンス計測関数

ex\_timesw, ex\_timerun, ex\_timeclear, ex\_timeread2, ex\_rh\_timeread, ex\_timeevent2, ex\_rh\_timeevent, ex\_timemode, ex\_rh\_performancesw, ex\_rh\_performanceclear, ex\_rh\_performanceread, ex\_rh\_performanceset

#### Trace function

1. Enable / disable the trace function

ex\_trcsw

2) Trace start / stop

ex\_trcrun

3) Set / reference trace operation status

ex\_trcmode

4) Clear trace data

ex\_trcclear

5) Set / reference / delete trace event conditions

ex\_trcevent

6) Trace data search condition setting / reference

ex\_trcsearch

7) Moving the trace pointer

ex\_trcseek

8) Set / reference delay value

ex\_trcdelay

9) Read trace data

ex\_trcread

10) Set / reference trace factor

ex\_m\_trceventsel2

11) Trace hardware buffer control settings

ex\_m\_trcbufcont

ex\_rh\_trcbufcont

12) Obtain information necessary for reading raw trace data

ex\_trcrawmeminfo

13) Trace raw data read

ex\_trcrawmemread

14) Hardware adjustment function related to trace data acquisition

ex\_trchwtune

15) Refers to hardware event number information corresponding to a single event number at trace startup

ex\_rh\_trchwevncopeinfo

16) Trace module (TCU) adjustment function

ex\_rh\_trctune

17) Set / reference trace filter conditions

ex\_rh\_trcfilter

[Glossary]

The following is an explanation of terms described in relation to trace.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Terms in the description | Explanation | Remarks |
| 1 | Enable / Disable | Indicates when the operation of the trace function is specified.  For example, when the trace function is valid, it means activation of the trace function (acquisition / recording of trace data).  When the trace function is disabled, it means that the trace function is not started (trace data is not acquired / recorded). | The trace function can be enabled / disabled if a trace module (TCU) is implemented in the device, but if not implemented, the trace function is always disabled. Note that if a trace module (TCU) is mounted on the device, a trace-related I / F can be issued even if the trace function is disabled. |
| 2 | Start / stop | Refers to the case of trace module (TCU) and EXEC operation.  For example, when tracing is started, the trace module (TCU) can acquire / record trace data, and the EXEC cannot read the trace data from the trace memory.  When the trace is stopped, the trace module (TCU) cannot acquire / record the trace data, and the EXEC can read the trace data from the trace memory. | When tracing is enabled, you can switch between starting and stopping trace. When the trace is invalid, it cannot be changed to the trace activation state. |
| 3 | start end | This refers to the timing of actually acquiring / recording trace data while the trace module (TCU) is running.  For example, the section trace start condition refers to the timing at which the trace module (TCU) starts acquiring trace data.  The term "section trace end condition" refers to the timing at which the trace module (TCU) ends acquisition of trace data.  In the case of qualifying trace, it refers to the range (from the start timing to the end timing) where the trace module (TCU) acquires trace data. | － |
| 4 | Acquisition mode | This refers to the process of generating trace information packets from each core (PE, local RAM) and slaves (global RAM / peripheral IO) and temporarily storing them in the trace buffer. The acquisition modes include “acquisition type”, “acquisition condition”, and “acquisition information”. The acquisition type includes a non-real-time trace mode and a real-time trace mode. The acquisition conditions include a condition trace (section trace / qualify trace) and a full trace. The acquired information includes a branch trace, a data access trace, and a soft trace. | － |
| 5 | Recording mode | This means that the trace information packet temporarily stored in the trace buffer is stored in a built-in trace memory or output to the outside and stored in a trace memory inside the emulator. The recording modes include a “ring mode”, a “full stop mode”, a “full break mode”, a “delay stop mode”, and a “delay break mode”. | "Delay break mode" is not supported by OCD emulator. |

[Setting target for each I / F]

The target to be set for the trace-related I / F is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| No | I/F名 | 設定対象 | 備考 |
| 1 | ex\_trcsw | Acquisition side |  |
| 2 | ex\_trcrun | Recording side |  |
| 3 | ex\_trcmode | Acquisition side | 「Recording side」There are also parameters (tcycle, tmode2, fullstop) classified as「Acquisition side」There is a parameter (tmode) that is classified as, and both are set collectively, so the setting target is “Acquisition side”. |
| 4 | ex\_trcclear | Recording side |  |
| 5 | ex\_trcevent | Acquisition side |  |
| 6 | ex\_trcdelay | Recording side |  |
| 7 | ex\_rh\_trceventsel2 | Acquisition side |  |
| 8 | ex\_rh\_trcbufcont | Acquisition side  Recording side | Acquisition side： When trc\_buftype = EX\_TRCLTB\_CPUTCMBUFF, EX\_TRCLTB\_SLAVE  Recording side： trc\_buftype = other than above |
| 9 | ex\_trcrawmeminfo | Recording side |  |
| 10 | ex\_trcrawmemread | Recording side |  |
| 11 | ex\_rh\_trchwevncopeinfo | Recording side |  |
| 12 | ex\_rh\_trctune | Acquisition side  Recording side | Acquisition side： When trc\_tunetype = EX\_TRC\_CPUSYNCMSG, EX\_TRC\_CPUWATCHMSG, EX\_TRC\_CPUMSGCHG, EX\_TRC\_CPUDBMSG, EX\_WPTSMSG  Recording side： trc\_tunetype = other than above |
| 13 | ex\_rh\_trcfilter | Acquisition side |  |

Acquisition side： I / F set for the currently selected physical CPU (PE) or physical slave resource. If the selected acquisition side trace function (TCU) is not implemented, the I / F returns EX\_DE\_SELTRCNOMOUNT. However, if the trace function is not implemented, I / F returns EX\_DE\_TRCNOMOUNT. Refer to ex\_rh\_getenv for the details of TCU implementation information on Acquisition side.

Recording side： I / F set for the recording side of the trace function. If no trace recording function (internal trace, external trace) is implemented, I / F returns EX\_DE\_TRCNOMOUNT. If the device does not have any of the PE internal master trace functions (TCU\_CPU, TCU\_TCM, etc.) or slave trace functions (TCU\_GRM, TCU\_EMS, TCU\_GVC, etc.), the I / O F returns EX\_DE\_TRCNOMOUNT. Refer to ex\_rh\_getenv for details of implementation information.

If either the internal side trace or the external side trace of the recording side is mounted on both the above “Acquisition side” / “Recording side” and neither is selected, the I / F returns EX\_DE\_TRCDISABLE.

[Handling of request setting at error response]  
When each trace function is issued and responds with an error, the setting value requested at the time of issuance is discarded, and the EXEC internal information and the setting value of the trace-related DCU register are not updated.

[About acquiring and releasing events when accessing data]

When no data access event condition is set in the qualifying trace condition and the debugger requests "Acquire data access" as the acquisition condition, it is necessary to use one TEU access event.

At the time requested by the debugger under the above conditions, the TEU access event 1ch is acquired as the EXEC internal processing. After the acquisition, TEU access events that can be used by the debugger due to other break events are reduced by one channel.  
When any of the above condition requests is released, the TEU access event 1ch acquired as the EXEC internal processing is released. After release, TEU access events that can be used by the debugger due to other break events increase by one channel.

(In the RH850, it is necessary to acquire 1ch of TEU in order to acquire the data access trace message due to the difference in the specification of the TCU module from the V850. In order to do this, the RH850 EXEC controls to automatically assign the data access TEU to 1ch.)

If no data access event condition is set as a qualifying trace condition, and the debugger requests "Acquire data access" as an acquisition condition, all TEU access events such as break events If used, the TEU access event cannot be acquired in the EXEC internal processing. In that case, an error response is sent to the debugger.

When one or more data access event conditions are set as qualifying trace conditions, TEU access events are not acquired even if the debugger requests "acquire data access" as the acquisition condition.

In addition, when no data access event condition is set in the qualifying trace condition and when "Acquire data access" is set, the data access event condition is set as the qualifying trace condition. At this time, the 1ch access event acquired in the above-described EXEC internal processing is released by the EXEC internal processing.

The relationship between trace mode and data access acquisition is as shown in the table below.  
(It is assumed that the trace function is enabled.)

Table 3 14 Relationship between trace mode and data access acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data access  (ex\_rh\_trceventsel2) | Trace mode  (ex\_trcmode) | Qualify trace conditions for data access events  (ex\_trcevent) | Acquisition of TEU access event 1ch | Acquisition of data access trace data |
| Do not get | All traces | 0 | do not do | Do not get data access |
| 1 or more |
| Section trace | 0 |
| 1 or more |
| Qualify Trace | 0 |
| 1 or more |
| get | All traces | 0 | Do | Get all data access for all time |
| 1 or more | do not do | Get only the data access set for the event of the entire period |
| Section trace | 0 | Do | Get all data access during section ON |
| 1 or more | do not do | Acquire only the data access set for the event during the section ON |
| Qualify Trace | 0 | Do | Get all data access during the qualifying period |
| 1 or more | do not do | Acquire only the data access set for the event during the validity period |

[About analysis and use of time stamp information]

In PFC1A and F1A v1, the time stamp information included in each packet is not added. Also, it does not record a time stamp packet generated when the time stamp counter overflows.  
Therefore, the time stamp information cannot be analyzed and used.  
In PFC1B and F1A v2 and later, time stamp information is added.

[Precautions when supporting three-tier virtual machines (G3 generation)]

In the trace function, there are no items that need to be set when supporting a three-layer virtual machine. When operating in a three-tier virtual machine, it automatically responds to the acquisition of trace messages for virtual machines and hardware threads.

[Precautions when supporting virtual machines (G4 generation)]

Trace output can be specified for each CPU operation mode. The specification of the trace output is linked with the specification of the debug target (unit\_type = EX\_UNIT\_HEGESEL of ex\_socunitinfo2). The setting of trace output is reflected only in the CPU trace, and the slave trace is not affected.  
In addition, trace data of CPU operation mode switching information (guest ← → host) can be output. Whether to output CPU operation mode switching information is specified by ex\_rh\_trceventsel2.

[Notes on slave trace]

If a break occurs during DMA transfer when all of the following conditions are met, a trace message may be recorded after the break. At this time, unintended overwriting may occur depending on the amount of messages recorded after the break.

1. Select internal trace
2. The recording mode is set to ring mode (EX\_NON\_STOP) and full break mode (EX\_FULL\_BRK).
3. Set <TCU\_GVC / TCU\_KVC / TCU\_AXI / TCU\_DMA> to acquisition, or set <GRAM / CRAM>, LRAM as DMA transfer source / destination, and set <TCU\_GRM / TCU\_CRM>, <TCU\_LRM> to acquisition. ing. Note that this does not occur when <TCU\_EMS> is set for acquisition.

[Notes on SPID filtering]

When performing filtering by SPID, acquire the trace of the enabled SPID regardless of the debug target specified by ex\_socunitinfo2 (EX\_UNIT\_HEGESEL).

##### ex\_trcsw

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **／** | **TIM** | **／** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **／** | **TRC2** | **×** |

[Function]

Specify enable / disable of the trace function.

[Format]

int ex\_trcsw( USHORT tmemsw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | Target |
| USHORT | tmemsw | <I > | Trace function enabled / disabled | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| tmemsw | | |
|  | EX\_TRCMEM\_OFF | Trace function disabled |
|  | EX\_TRCMEM\_ON | Trace function enabled |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC not started

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_TRACEON Starting trace

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_ALREADY Already in the same state

EX\_PE\_EVNOVER Event setting number exceeded

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable for asynchronous debug mode because of non-target cores Not available for asynchronous debug mode because of non-target cores

[Description]

[Argument description]

(1)tmemsw

Specify enable / disable of the trace function (TCU). Default value is as follows.

TCU\_CPU: EX\_TRCMEM\_ON (However, EX\_TRCMEM\_OFF when TCU\_CPU is not mounted on the device)

Other than TCU\_CPU (slave trace)：EX\_TRCMEM\_OFF

[Description of operation]

Specify enable / disable of the trace function (TCU).

If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.

Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.

Whether or not the trace function (TCU) is installed can be referenced by ex\_rh\_getenv.

When no data access event condition is set in the qualify / trace conditions and the trace function is enabled with the trace acquisition condition set to "Acquire data access", the unused TEU If the access event does not exceed 1ch, an error response is returned and the trace function is disabled.

For details on consuming one channel of events as an internal process of the EXEC, refer to the trace function [Event acquisition and release at data access].

[Notes on multi-core and slave support]

(1) Set for the currently selected physical CPU (PE) or physical slave resource.

(2) Switching of physical CPU (PE) and physical slave resources

Use ex\_socunitinfo2 (EX\_UNIT\_COREPESEL / EX\_UNIT\_SLAVEIDSEL).

**[Notes for debuggers]**  
(1) PFC1A does not support the slave trace function.  
**(2) If this function is enabled for a physical CPU (PE) that is set not to acquire branch trace, data access trace, and soft trace, unnecessary trace data (such as timestamp synchronization messages ) May be acquired, and extra trace memory is consumed. To avoid this, disable the physical CPU (PE) that does not acquire trace data with this function.**  
(3) If parameters different from the last time are set, clear the trace data.  
**(4) Change the trace output type (external trace, internal trace) with ex\_socunitinfo2 (EX\_UNIT\_TRCSEL).**

##### ex\_trcrun

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **×** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **○** | **TRC2** | **×** |

[Function]

Specify start / stop of trace during execution of user program.

[Format]

int ex\_trcrun( USHORT trc\_sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | Target |
| USHORT | trc\_sw | <I > | Trace start / stop | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| trc\_sw | | |
|  | EX\_TRC\_RUN | Trace activation |
|  | EX\_TRC\_STOP | Stop trace |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC not started

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEMEMOFF Trace memory is disabled

EX\_SE\_USRPGMSTOP During user program break

EX\_SE\_TRACEON Trace activation is specified while trace activation is in progress

EX\_SE\_TRACEOFF Trace stop is specified while trace is stopped

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)trc\_sw

Specify start / stop of the trace during execution of the user program.

[Description of operation]

Specify start / stop of the trace during execution of the user program.

If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.

Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.

**Refer to ex\_rh\_getenv for the implementation of the trace function (TCU)**.

If the item of ex\_rh\_trceventsel2 is set to acquire the slave trace function, also start / stop the slave trace function.

Even if the trace operation is stopped by specifying the trace stop during the execution of the user program, the trace is started again at the time of re-execution after the break of the user program. Note that trace activation may not be performed again depending on the contents specified in ex\_socunitinfo2 (EX\_UNIT\_RUNCONTOPT).

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).  
(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)

An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

[Notes when the initial stop / standby mode debug environment is enabled]  
While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) This function can be issued when there is at least one physical CPU (PE) or slave resource that is validly specified by ex\_trcsw. If this function is issued without any physical CPU (PE) or slave resources specified as valid in ex\_trcsw, an error response is returned.

(2) Switching of physical CPU (PE) or slave resources

Use ex\_socunitinfo2 (EX\_UNIT\_COREPESEL / EX\_UNIT\_SLAVEIDSEL).

(3) Starts / stops the trace for all physical CPUs (PEs) or slave resources that are specified as valid in ex\_trcsw.

[Precautions for debuggers]

(1) PFC1A does not support the slave trace function.

##### ex\_trcmode

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **×** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

Set / reference trace operation conditions.

[Format]

int ex\_trcmode( USHORT func, EXTRMODE \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| EXTRMODE | \*cond | <IO> | Trace operation conditions | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | tcycle | <IO> | Trace cycle | ○ |
|  | USHORT | tmode | <IO> | Trace mode | ○ |
|  | USHORT | tmode2 | <IO> | Trace mode 2 | ○ |
|  | USHORT | fullstop | <IO> | Trace end mode | ○ |
|  | ULONG | port\_sel | <IO> | Unused (0 fixed) | -- |
|  | USHORT | tstamp\_mode | <IO> | Unused (0 fixed) | -- |
| }EXTRMODE | | |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| func | | | | |
|  | EX\_SET | | Setting | |
|  | EX\_REF | | reference | |
|  |  | |  | |
| tcycle | | | | |
|  | EX\_REALTIME\_MODE | | Real-time trace mode (default) | |
|  | EX\_FULL\_MODE | | Non real-time trace (complete trace) mode | |
|  |  | |  | |
| tmode | | | | |
|  | EX\_SECQUAL\_TRC | | Trace using section trace or qualify trace of ex\_trcevent | |
|  | EX\_ALL\_TRC | | All traces (default) | |
|  |  | |  | |
| tmode2 | | | | |
|  | Specify trace data recording conditions in bit units | | | |
|  | bit0 |  | | Specify storage method in trace memory  0　　　　　　　：Normal data mode  EX\_M\_RAWTRCMODE：Raw data mode  \* Not supported (EX\_M\_RAWTRCMODE fixed) |
|  | bit1 |  | | Specify whether to trace DMA  ※ Unused (0 fixed) |
|  | bit2-13 |  | | ※ Unused (0 fixed) |
|  | bit14 |  | | Specify whether to complete instruction at branch trace  ※ Unused (EX\_M\_NOCOMPLEMENT fixed)  EX\_M\_COMPLEMENT ：With instruction completion  EX\_M\_NOCOMPLEMENT：No instruction completion |
|  | bit15 |  | | ※ Unused (0 fixed) |
|  | | | | |

|  |  |  |
| --- | --- | --- |
| fullstop | | |
|  | EX\_NON\_STOP | Do not end tracing (default) |
|  | EX\_FULL\_STOP | End when trace memory is full |
|  | EX\_DLY\_STOP | End with delay event |
|  | EX\_FULL\_BRK | Break when trace memory is full |
|  | EX\_DLY\_BRK | Break on delay event (valid only when external trace is selected) |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC not started

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEON Starting trace

EX\_PE\_EVNOVER Event setting number exceeded

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_PE\_SELTRC Cannot be set for the selected trace resource

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_ASYNCMODE\_ENB Cannot be set / referenced / deleted while in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)func

Specify the operation mode.

EX\_SET：Set the trace operation condition specified by cond.

EX\_REF：Returns the currently set trace operation conditions to cond.

(2)tcycle

Specify the trace operation mode.

EX\_REALTIME\_MODE：Real-time trace mode

(A mode in which branching, data access, etc. are observed in real time during execution of the user program)

EX\_FULL\_MODE ：Non real-time trace mode

(A mode to observe all branches executed by the user program, data access, etc.)

(3)tmode

Specify the trace acquisition conditions.

EX\_SECQUAL\_TRC：Section trace or qualify trace

(Event conditions are set by ex\_trcevent)

EX\_ALL\_TRC ：All traces

(4)tmode2

Specifies the conditions for recording trace data in bit units.

bit0 ：EX\_M\_RAWTRCMODE (raw data mode) fixed

bit[4:2]：Not supported (fixed at 0)

bit14 ：EX\_M\_NOCOMPLEMENT (return only branch address without completion) fixed

Other ：Unused (0 fixed)

(5)fullstop

Specify the conditions for recording trace data.

EX\_NON\_STOP ：Even if the trace memory becomes full, the recording operation of the trace data is not terminated and the recording operation is continued. (Ring mode / Overwrite mode)

EX\_FULL\_STOP：When the trace memory becomes full, the recording operation of the trace data ends.

EX\_DLY\_STOP ：After acquiring the specified number of trace data after acquiring the watch point message due to the occurrence of the delay event, the recording operation of the trace data is terminated.

EX\_FULL\_BRK ：When the trace memory becomes full (full), the trace data recording operation is terminated, and a break request of the user program is issued.

EX\_DLY\_BRK ：When the specified number of trace data is acquired after the acquisition of the watch point message due to the occurrence of the delay event, the operation of recording the trace data is terminated, and a break request of the user program is issued.

[Description of operation]

Set / reference trace operation conditions, trace data recording conditions, and recording operation conditions.

If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.

Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.

When EX\_REALTIME\_MODE is specified in tcycle, overflow (lost) may occur. In that case, trace data relating to an error indicating that an overflow has occurred is stored in the trace memory.

When EX\_FULL\_MODE is specified for tcycle, the occurrence of overflow is reduced because the recording operation of trace data has priority over the execution of the user program.

A local trace buffer in the TCU that uses ex\_rh\_trcbufcont to request / release pending / executed instruction / data access execution in non-real-time trace mode and release the suspended state after a loss of trace data in real-time trace mode / It is necessary to prevent overflow by setting the number of stages in the global trace buffer.

When EX\_SECQUAL\_TRC is specified in tmode, the event condition specified in ex\_trcevent is set.

When both the section trace start / end event condition and the qualify event condition are set in ex\_trcevent, the acquisition / recording of trace data is performed as an AND condition of the section trace and the qualify trace.

In cases other than the above (only the section trace start / end condition event condition, only the qualify event condition, etc.), the trace data is acquired / recorded as the OR condition.

If EX\_ALL\_TRC is specified for tmode, the event condition specified by ex\_trcevent is ignored, and all trace data set to be acquired by item of ex\_rh\_trceventsel2 is recorded.

An error response is given during trace activation.

If trace slave resource is specified by ex\_socunitinfo2, qualify trace is not supported. (TCU restrictions)

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).

(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)

An error occurs if this function is called during execution of a user program while the E2 extended function is enabled. (EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

[Precautions when the initial stop / standby mode debug environment is enabled]

While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) Set for the currently selected physical CPU (PE) or slave resource.

(2) Switching of physical CPU (PE) or slave resources

Use ex\_socunitinfo2 (EX\_UNIT\_COREPESEL / EX\_UNIT\_SLAVEIDSEL).

[Precautions for debuggers]

(1) In this function, there are parameters that are set / referenced for each PE or slave resource, and parameters that are set / referenced in common, as shown below.

Parameter set / referenced for each PE or each slave resource: tmode

Common parameters: tcycle, tmode2, fullstop

(2) If parameters different from the last time are set, the trace data is cleared.

(3) When the trace output type (external trace, internal trace) is changed, the setting value is initialized. Note that the trace output type (external trace, internal trace) can be changed with ex\_socunitinfo2 (EX\_UNIT\_TRCSEL).

(4) If an external trace is selected when the IE850A emulator is connected and the recording operation (fullstop) is set to EX\_FULL\_BRK, the trace message may exceed the maximum capacity of the trace memory depending on the timing at which a full break occurs. . In this case, a trace message exceeding the capacity is not recorded, and ex\_getstatus es\_trc\_overfullbreak = 1.

(5) If the recording operation (fullstop) is set to EX\_DLY\_BRK when the external trace is selected when the IE850A emulator is connected, more trace messages may be output than the value set by ex\_trcdelay after a delay event occurs. For this reason, if ex\_getstatus es\_trc\_wraparound = 1 (wraparound state) at the time of a break, more trace messages than expected may be overwritten.

(6) The recording operation (fullstop) that can be set for each emulator is shown in the table below.

[Notes on asynchronous debug mode]

(1) When the asynchronous debug mode is set, only the target core breaks in the full break mode or the delay break mode. See section 2.7 for details on asynchronous break mode.

(2) The table below shows the recording operations (fullstop) that can be set for each emulator when the asynchronous debug mode is set.

Table 3 15 Whether fullstop parameter can be set when internal trace is selected

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| emulator | | E1/E20 | | E2 | | IE850 | | IE850+SP | | IE850A | |
| Synchronous debug mode | | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous |
| fullstop | EX\_NON\_STOP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| EX\_FULL\_STOP | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 |
| EX\_DLY\_STOP | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 | ○※3 |
| EX\_FULL\_BRK | ○ | ×※2 | ○ | ×※2 | ○ | ×※2 | ○ | ×※2 | ○ | ×※2 |
| EX\_DLY\_BRK | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 | ×※1 |

○: Available, ×: Not available

\* 1 Error response (EX\_UNIT\_TRCSEL)

\* 2 Error response (EX\_SE\_ASYNCMODE\_ENB)

\* 3 Cannot be set when tcycle = EX\_FULL\_MODE. (No error response)

Table 3 16 Whether fullstop parameter can be set when external trace is selected

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| emulator | | E1/E20 | | E2 | | IE850 | | IE850+SP | | IE850A | |
| Synchronous debug mode | | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous | Same period | asynchronous |
| fullstop | EX\_NON\_STOP | － | － | － | － | ○ | ○ | ○ | ○ | ○ | ○ |
| EX\_FULL\_STOP | － | － | － | － | ○ | ○ | ○ | ○ | ○ | ○ |
| EX\_DLY\_STOP | － | － | － | － | ○ | ○ | ×※1 | ×※1 | ○ | ○ |
| EX\_FULL\_BRK | － | － | － | － | ○ | ○ | ×※1 | ×※1 | ○ | ○ |
| EX\_DLY\_BRK | － | － | － | － | ○ | ○ | ×※1 | ×※1 | ○ | ○ |

○: Available, ×: Not available,-: No status

\* 1 Error response (EX\_UNIT\_TRCSEL)

##### ex\_trcclear

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **×** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

Clear trace data.

[Format]

int ex\_trcclear( void )

[Arguments]

None

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC not started

EX\_SE\_TRACEON Starting trace

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Description of operation]

Initialize the trace data (discard the acquired trace data). At the same time, the time stamp is also initialized.

By initialization, status flags such as trace data full and trace data lost are also initialized.

If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.

Note that an error response is returned during trace activation because initialization cannot be performed.  
Also, depending on the device, the trace function (TCU) may not be implemented. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.  
Refer to ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) for the implementation of trace function (TCU).

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).

(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)

An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

[Precautions when the initial stop / standby mode debug environment is enabled]

While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) Initialize in common regardless of the physical CPU (PE) and slave resources.

[Notes on asynchronous debug mode]

(1) Even if the debug target core is running in the asynchronous debug mode, it can be called if the trace is stopped.

##### ex\_trcevent

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **○** | **TRC2** | **×** |

[Function]

Set / reference / delete trace event conditions.

[Format]

int ex\_trcevent( USHORT func, EXTREVENT \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | Target |
| USHORT | func | <I > | Operation mode | ○ |
| EXTREVENT | \*cond | <IO> | Trace event condition | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <I > | Trace conditions | ○ |
|  | USHORT | num | <IO> | Event setting number | ○ |
|  | USHORT | \*event | <IO> | Event number (multiple numbers allowed) | ○ |
| }EXTREVENT | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | Setting |
|  | EX\_REF | reference |
|  | EX\_DEL | Delete |
|  |  |  |
| mode | | |
|  | <Condition type of trace event> | |
|  | EX\_RH\_SEC\_ENA\_EVENT | Section trace start condition |
|  | EX\_RH\_SEC\_DIS\_EVENT | Section trace termination condition |
|  | EX\_RH\_QUAL\_EVENT | Qualify trace conditions |
|  | EX\_RH\_DLY\_EVENT | Delay trigger condition |
|  |  |  |
| num | | |
|  | <1 - > |  |
|  |  |  |
| event | | |
|  | Buffer pointer containing num event numbers | |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEON Starting trace

EX\_SE\_NONEVNNO An event number for which no condition has been set was operated

EX\_PE\_INVEVNCOND Contrary to restrictions on event number setting conditions

EX\_PE\_EVNOVER Event setting number exceeded

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_PE\_SELTRC Cannot be set for the selected trace resource

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_EVNDELETE Unable to delete event

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_TRACEON Starting trace

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)func

Specify the operation mode.

EX\_SET：Set the event numbers stored in num events in the trace condition specified by mode.

EX\_REF：Returns the event number currently set in the trace condition of mode.

EX\_DEL：Deletes num event numbers stored in event from the trace conditions specified in mode.

(2)mode

Specify the trace condition of the event number to be set / referenced / deleted.

(3)num

Specify the number of event numbers stored in event.

(4)event

Specify the buffer address that stores num event numbers.  
The event number specifies the event number specified by ex\_rh\_evncond and ex\_rh\_evnintgcond.

[Description of operation]

Set / reference / delete the start / end of the section trace and the event number that generates the qualify trace.  
If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.  
Note that some devices do not have the trace function (TCU), and in this case, return an error (EX\_DE\_SELTRCNOMOUNT). For details on returning this error, refer to [Setting target for each I / F] of the trace function.

At the time of setting, if an already set event number is specified, it is ignored and processing continues.  
If an event number other than those specified in ex\_rh\_evncond and ex\_rh\_evnintgcond (event number with unset conditions) is specified, an error (EX\_SE\_NONEVNNO) is returned.  
When mode = EX\_RH\_QUAL\_EVENT, only the event number set in the TEU\_CPU event condition and TEU slave event condition in ex\_rh\_evncond can be specified. If any other is specified, an error (EX\_PE\_INVEVNCOND) is returned.  
Also, even if the event number specified by ex\_rh\_evncond or ex\_rh\_evnintgcond, an error (EX\_PE\_INVEVNCOND) is returned even if the physical CPU (PE) to be set is different.

If tmode = EX\_SECQUAL\_TRC is not set in ex\_trcmode, the specified event number will not be valid.  
However, when the event number set in the data access event condition is specified when mode = EX\_RH\_QUAL\_EVENT, it is valid regardless of the t\_mode of ex\_trcmode.  
Attempting to delete all data access event conditions when "Acquire data access" is set, if there is no remaining event for one or more channels, an error response (EX\_SE\_EVNDELETE) is returned and the event cannot be deleted. In this case, delete the data after setting "No data access".  
At the time of reference, if no event number is set in the specified trace condition, 0 is returned to num and no error response is returned.

At the time of deletion, if an event number other than those specified in ex\_rh\_evncond and ex\_rh\_evnintgcond (event number with unset conditions) is specified, ignore it and continue the process.  
If setting / deletion is specified during trace startup, an error (EX\_SE\_TRACEON) is returned.

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).  
(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

[Notes when the initial stop / standby mode debug environment is enabled]  
While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) Set for the currently selected physical CPU (PE) or slave resource.

(2) Switching of the physical CPU (PE) or slave resource is performed by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL / EX\_UNIT\_SLAVEIDSEL).

[Notes for debugger developers]

(1) If parameters different from the last time are set, the trace data is cleared.

(2) When the trace output type (external trace, internal trace) is changed, the setting value is initialized. Note that the trace output type is changed using ex\_socunitinfo2 (EX\_UNIT\_TRCSEL).

(3) When the single event set in the qualifying trace condition is changed from ex. The trace data is not cleared as when the acquisition operation was changed in the system I / F.

(4) When the status condition (ex\_rh\_evncond-stat) of the single event set in the qualifying trace condition is as follows, the data condition (ex\_rh\_evncond-data) of the read data is ignored. A read access in which all conditions except the data condition are detected is output as a trace. In the case of write data, it is not ignored.  
ex\_rh\_evncond-stat = EX\_RWP\_STS, EX\_RP\_STS, EX\_RWPND\_STS, EX\_RPND\_STS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_trcsearch **[This function is not supported]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

トレース・データの検索条件の設定/参照を行う。

[Format]

int ex\_trcsearch( USHORT func, EXTRSEARCHCON \*sd )

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_trcseek **[This function is not supported]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

トレース・ポインタを移動する。

[Format]

int ex\_trcseek( EXTRSEEK \*sd, ULONG \*tp )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXTRCSEEK | \*sd | <--> | ---------- | -- |
| ULONG | \*tp | <--> | ---------- | -- |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <--> | ---------- | -- |
|  | USHORT | origin | <--> | ---------- | -- |
|  | ULONG | count | <--> | ---------- | -- |
|  | USHORT | direct | <--> | ---------- | -- |
| }EXTRMODE | | |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

##### ex\_trcdelay

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **×** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

ディレイ値を指定/参照する。

[Format]

int ex\_trcdelay( USHORT func, ULONG \*delay )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| ULONG | \*delay | <IO> | ディレイ値 | ○ |
|  | | | | |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEON Starting trace

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_PE\_SELTRC Cannot be set for the selected trace resource

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)func

ディレイ値を設定/参照する。

func=EX\_SET：delayに指定されたディレイ値を設定する。

func=EX\_REF：現在設定しているディレイ値をdelayに返す。

(2)delay

設定/参照するディレイ値を指定する。

func=EX\_SET：delayに指定されたディレイ値の設定を行う。

func=EX\_REF：現在設定しているディレイ値を返す。

[動作説明]

トレースを停止するまでのディレイ値を設定/参照する。

トレース出力タイプに内蔵トレースもしくは外部トレースのどちらも選択されていなかった場合は、エラー(EX\_DE\_TRCDISABLE)を返す。

なお、デバイスによってはトレース機能(TCU)が未実装の場合もあり、この場合はエラーを返す。このエラーを返すときの詳細は、トレース関数の[各I/Fごとの設定対象]を参照。

ディレイ値とは、ディレイイベント条件が成立してからトレース停止するまでにトレースメモリに格納するフレーム数である。

トレース起動中は、エラー応答する。

ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。

(次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に設定可能とする)

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[マルチコア時およびスレーブ対応時の注意点]

(1)トレースメモリに関する指定であるため、物理CPU(PE)および、スレーブ資源の区別なく共通設定となる。

[デバッガ開発者向け注意事項]

1. フレームは、トレース・データの解析が行える単位(トレースメッセージ)ではなく、トレースメモリへの格納単位である。また、1トレース・データの解析に必要なフレーム数は、トレースメッセージによって可変であるため、最終フレームがトレースメッセージの区切りのフレームでない場合、解析ができない可能性がある。

(2)前回と異なるパラメータを設定した場合、トレース・データのクリアを行う。

(3)トレース出力タイプに外部トレースを選択していて、SPボードを接続しているときに、ディレイ値を設定した場合にはエラー(EX\_PE\_SELTRC)を返す。

(4)トレース出力タイプ(外部トレース、内蔵トレース)を変更したとき、設定値は初期化される。なお、トレース出力タイプの変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

(5)トレース出力先と接続エミュレータ種別に対して、delayパラメータの設定可能値は異なる。delayパラメータの設定可能値は、表 3‑17 を参照。設定可能な範囲外を設定した場合は、エラー(EX\_PE\_DATA)を返す。

表 3‑17 delayパラメータの設定可能値

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | エミュレータ  種別 | トレース  メモリ容量 | delayパラメータの設定可能値 | | 備考 |
| 最小値 | 最大値 |
| 内蔵トレース | － | 32kB | 00000000h | 000007FFh |  |
| － | 64kB | 00000000h | 00000FFFh |  |
| 外部トレース | IE850 | 16MB | 00000009h | 001FFFFEh※1 | 偶数値のみ設定可能、奇数値を設定された場合は、+1した値が設定される |
| IE850A | 8GB | 00000000h | 3FFFFFFFh※1 |  |

※1 ex\_setenvのse.trcmem\_sizeパラメータの設定により異なる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_trcread **[This function is not supported]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

トレース・データを読み出す

[Format]

int ex\_trcread( EXTRREAD \*rd, ULONG \*tp )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXTRREAD | \*rd | <--> | ---------- | -- |
| ULONG | \*tp | <--> | ---------- | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <--> | ---------- | -- |
|  | USHORT | direct | <--> | ---------- | -- |
|  | ULONG | num | <--> | ---------- | -- |
|  | ULONG | rnum | <--> | ---------- | -- |
|  | ULONG | dsize | <--> | ---------- | -- |
|  | ULONG | \*data | <--> | ---------- | -- |
| }EXTRREAD | | |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_m\_trceventsel2 **[This function is not supported]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

トレースイベント要因やメッセージ選択等

[Format]

int ex\_m\_trceventsel2( USHORT func, ULONG \*item, USHORT \*trc\_num, EXDATTRC2 \*trc\_cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <--> | ---------- | -- |
| ULONG | \*item | <--> | ---------- | -- |
| USHORT | \*trc\_num | <--> | ---------- | -- |
| EXDATTRC2 | \*trc\_cond | <--> | ---------- | -- |
|  |  |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

##### ex\_rh\_trceventsel2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **×** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

Set / refer to whether or not to acquire trace data and the type of trace data to be acquired.

[Format]

int ex\_rh\_trceventsel2( USHORT func, ULONG \*item, USHORT \*trc\_num, EXRHDATTRC2 \*trc\_cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| ULONG | \*item | <IO> | Specification of trace event cause and message selection | ○ |
| USHORT | \*trc\_num | <--> | Not used (may be NULL) | -- |
| EXRHDATTRC2 | \*trc\_cond | <--> | Not used (may be NULL) | -- |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| func | | | | |
|  | EX\_SET | | | Configuration |
|  | EX\_REF | | | reference |
| \*item | | | | |
|  | Set conditions for recording trace data in bit units | | | |
|  | bit0 |  | Whether to acquire trace data for data read access output from the slave resource specified in ex\_socunitinfo2 0: Do not acquire (Default) EX\_RH\_EVN\_SLVRENB: Get | |
|  | bit1 |  | Whether to acquire trace data for data write access output from the slave resource specified by ex\_socunitinfo2 0: Do not acquire (Default) EX\_RH\_EVN\_SLVWENB: Get | |
|  | bit2 |  | Whether to acquire the timestamp trace data with the virtualization information added when switching the CPU operation mode output from the CPU trace (TCU\_CPU)  Only G4 generation. Fixed to 0 for G3 generation.)  0 ：Do not get (Default)  EX\_RH\_EVN\_VMTRANSENB：get | |
|  | bit[8:3] |  | Unused (fixed at 0) | |
|  | bit9 |  | Acquisition of trace data of data read access output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_RDATALL: Get (Default) | |
|  | bit10 |  | Whether to acquire trace data for data write access output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_WDATALL: Get (Default) | |
|  |  |  |  | |
|  | bit11 |  | Whether to acquire trace data of branch instruction output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_BRALL: Get (Default) | |
|  | bit[15:12] |  | Specify the trace data output type when accessing data output from the physical CPU (PE) or slave resource specified in ex\_socunitinfo2  ・When CPU trace is selected (corresponds to DMT [2: 0] bits in TRC\_CTRL register) When the relevant PE of ex\_rh\_getenv-funcinfo [3] = 0  EX\_RH\_EVN\_CPUDTM1\_1 ：See [Operation] (Default)  EX\_RH\_EVN\_CPUDTM2\_2 ：See [Operation]  EX\_RH\_EVN\_CPUDTM1\_35：See [Operation]  EX\_RH\_EVN\_CPUDTM2\_45：See [Operation]  EX\_RH\_EVN\_CPUDTM3\_3 ：See [Operation]  EX\_RH\_EVN\_CPUDTM4\_4 ：See [Operation]  When the relevant PE of ex\_rh\_getenv-funcinfo [3] = 1  EX\_RH\_EVN\_CPUDTM35\_35 ：See [Operation] (Default)  EX\_RH\_EVN\_CPUDTM45\_45 ：See [Operation]  EX\_RH\_EVN\_CPUDTM3\_3 ：See [Operation]  EX\_RH\_EVN\_CPUDTM4\_4 ：See [Operation]  When the relevant PE of ex\_rh\_getenv-funcinfo [3] = 2  EX\_RH\_EVN\_CPUDTM1\_1 ：See [Operation] (Default)  EX\_RH\_EVN\_CPUDTM2\_2 ：See [Operation]  EX\_RH\_EVN\_CPUDTM35\_35：See [Operation]  EX\_RH\_EVN\_CPUDTM45\_45：See [Operation]  EX\_RH\_EVN\_CPUDTM3\_3 ：See [Operation]  EX\_RH\_EVN\_CPUDTM4\_4 ：See [Operation]  ・When TCU\_EMS of slave trace is selected  (Corresponds to the DMT bit in the TRE\_CTRL register)  EX\_RH\_EVN\_EMSDTM1\_1 ：See [Operation](Default)  EX\_RH\_EVN\_EMSDTM3\_3 ：See [Operation]  ・When TCU\_GRM / TCU\_CRM of slave trace is selected  (Corresponds to the DMT bit in the TRG\_CTRL register)  EX\_RH\_EVN\_GRMDTM1\_1 ：See [Operation](Default)  EX\_RH\_EVN\_GRMDTM3\_3 ：See [Operation]  ・When TCU\_LRM of slave trace is selected  (Corresponds to the DMT bit in the TRT\_CTRL register)  EX\_RH\_EVN\_LRMDTM1\_3：See [Operation](Default)  EX\_RH\_EVN\_LRMDTM3\_3：See [Operation]  ・When TCU\_GVC / TCU\_KVC / TCU\_AXI of slave trace is selected  (Corresponds to the DMT bit in the TRE\_CTRL register)  EX\_RH\_EVN\_GVCDTM1\_35：See [Operation](Default)  EX\_RH\_EVN\_GVCDTM3\_3 ：See [Operation]  ・When TCU\_DMA of slave trace is selected  (Corresponds to the DMT bit in the TRE\_CTRL register)  EX\_RH\_EVN\_DMADTM35\_35：See [Operation](Default)  EX\_RH\_EVN\_DMADTM3\_3　：See [Operation] | |
|  |  |  |  | |
|  | bit16 |  | Invalidation of trace data acquisition for data access based on stack pointer (SP, R3) output from CPU trace (TCU\_CPU)  0: Acquisition valid (Default) EX\_RH\_EVN\_SFTSPD: Acquisition invalid | |
|  |  |  |  | |
|  | **bit17** |  | **Specify the output type of soft trace data (DBTAG instruction) output from CPU trace (TCU\_CPU)  EX\_RH\_EVN\_SFTSMT2: Output only immediate value EX\_RH\_EVN\_SFTSMT3: Output immediate value + execution PC (Default)** | |
|  | **bit18** |  | **Specifying the output type of the soft trace data (DBPUSH instruction) output from the CPU trace (TCU\_CPU)  EX\_RH\_EVN\_SFTSMT4: Output general-purpose register number + general-purpose register value EX\_RH\_EVN\_SFTSMT5: Output general-purpose register number + general-purpose register value + execution PC value (Default)** | |
|  | **bit19** |  | **Whether to acquire soft trace data (DBCP) output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_SFTDBCP: Get (Default)** | |
|  | **bit20** |  | **Whether to acquire soft trace data (DBPUSH) output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_SFTDBPUSH: Get (Default)** | |
|  | **bit21** |  | **Whether to acquire soft trace data (DBTAG) output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_SFTDBTAG: Get (Default)** | |
|  | **bit22** |  | **Enable / disable designation of time stamp function for internal trace  0: Disable time stamp function (Default) EX\_RH\_EVN\_TIMESTAMPENA: Enable time stamp function \* If EX\_RH\_EVN\_TIMESTAMPENA is set with unsupported devices fixed at 0, an error (EX\_PE\_SELTRC) is returned. Whether the time stamp function is supported is as follows.    PFC1A, F1L (v1): Not supported    PFC1B, F1L (v2) or later: Supported** | |
|  |  |  |  | |
|  | bit23 |  | ex\_rh\_getenv-funcinfo [0] -b3 = 0 Unused (fixed at 0) \* For devices equipped with both TCU\_GRM / TCU\_CRM and TCU\_GVC / TCU\_KVC / TCU\_AXI / TCU\_DMA, trace messages can be obtained simultaneously from both resources.  ex\_rh\_getenv-funcinfo [0] -b3 = 1 and slave trace TCU\_GRM / TCU\_CRM selected 0: Disable TCU\_GRM / TCU\_CRM, enable TCU\_GVC / TCU\_KVC / TCU\_AXI / TCU\_DMA (Default) 1: Enable TCU\_GRM / TCU\_CRM, disable TCU\_GVC / TCU\_KVC / TCU\_AXI / TCU\_DMA \* The trace setting for the disabled slave resource is valid, but no trace message is obtained. \* TCU\_GRM and TCU\_CRM are implemented exclusively. \* TCU\_GVC, TCU\_KVC, TCU\_AXI, TCU\_DMA are implemented exclusively. | |
|  | **bit24** |  | **Soft trace data (DBCP) output from CPU trace (TCU\_CPU) is always obtained regardless of the qualify / section subswitch status  0: Always disabled (Default) EX\_RH\_EVN\_SFTDBCP\_ALWAYS: Always valid (\* When this bit is set to EX\_RH\_EVN\_SFTDBCP\_ALWAYS, it is valid only when bit19 = EX\_RH\_EVN\_SFTDBCP, and when bit19 = 0, this setting is ignored.)** | |
|  | **bit25** |  | **Soft trace data (DBPUSH) output from CPU trace (TCU\_CPU) is always acquired regardless of the status of qualify / section subswitch  0: Always disabled (Default) EX\_RH\_EVN\_SFTDBPUSH\_ALWAYS: Always available (\* When this bit is set to EX\_RH\_EVN\_SFTDBPUSH\_ALWAYS, it is valid only when bit20 = EX\_RH\_EVN\_SFTDBPUSH, and this bit is ignored when bit20 = 0)** | |
|  | **bit26** |  | **Soft trace data (DBTAG) output from CPU trace (TCU\_CPU) is always obtained regardless of the status of qualify / section subswitch  0: Always disabled (Default) EX\_RH\_EVN\_SFTDBTAG\_ALWAYS: Always valid (\* When this bit is set to EX\_RH\_EVN\_SFTDBTAG\_ALWAYS, it is valid only when bit21 = EX\_RH\_EVN\_SFTDBTAG, and this bit is ignored when bit21 = 0)** | |
|  | bit[30:27] |  | Unused (0 fixed) | |
|  | bit31 |  | Acquisition of trace data of data access output from CPU trace (TCU\_CPU)  0: Do not acquire EX\_RH\_EVN\_DATAALL: Get (Default) \* When this bit is set to EX\_RH\_EVN\_DATAALL, the settings of bits [10: 9] are invalid. When this bit is set to 0, the settings of bits [10: 9] are valid | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEON Starting trace

EX\_DE\_M\_TRCMEMACCFAILED Trace memory access failed

EX\_PE\_EVNOVER Event setting number exceeded

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_PE\_SELTRC Cannot be set for the selected trace resource

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Parameter description]  
(1) func  
Set / reference the trace event factor and the output trace data type.  
func = EX\_SET: Sets the information specified in item.  
func = EX\_REF: Returns the currently set information to item

1. item  
   Specifies the conditions for recording trace data in bit units.  
   0: The condition of the corresponding bit is invalid  
   1: Setting: The condition of the corresponding bit is valid

[Operation description]  
Set / refer to the conditions for recording trace data.  
The relationship between the setting of bit [15:12] of item and the output type of trace data to be recorded is as follows

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Set value | Output type during data write access | | Output type during data read access | | Output of program correlation message (\* 1) |
| Address information | Data information | Address information | Data information |
| EX\_RH\_EVN\_CPUDTM1\_1  EX\_RH\_EVN\_EMSDTM1\_1  EX\_RH\_EVN\_GRMDTM1\_1 | ○ | ○ | ○ | ○ | × |
| EX\_RH\_EVN\_CPUDTM2\_2 | ○ |
| EX\_RH\_EVN\_LRMDTM1\_3 | △ | × | × |
| EX\_RH\_EVN\_CPUDTM1\_35  EX\_RH\_EVN\_GVCDTM1\_35 | ▲ | ▲ | × |
| EX\_RH\_EVN\_CPUDTM2\_45 | ○ |
| EX\_RH\_EVN\_CPUDTM35\_35  EX\_RH\_EVN\_DMADTM35\_35 | ▲ | ▲ | × |
| EX\_RH\_EVN\_CPUDTM45\_45 | ○ |
| EX\_RH\_EVN\_CPUDTM3\_3  EX\_RH\_EVN\_LRMDTM3\_3  EX\_RH\_EVN\_EMSDTM3\_3  EX\_RH\_EVN\_GRMDTM3\_3  EX\_RH\_EVN\_GVCDTM3\_3  EX\_RH\_EVN\_DMADTM3\_3 | △ | × | △ | × | × |
| EX\_RH\_EVN\_CPUDTM4\_4 | ○ |

○：Applicable information is output to the same trace data

△：Applicable information is output to separate trace data

▲：Applicable information is output to separate trace data (association of each information uses access ID information included in trace data)

×：The corresponding information is not output to the trace data.

\* 1: Trace data that outputs the absolute address information of the instruction that executed data access (output separately from trace data at data access)

If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.

Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.  
Also, if "Acquire data access" is set with this function when no data access event condition is set in the qualifying trace condition, if there is no unused TEU access event of 1ch or more, Respond with an error.

For details on consuming one channel of events as an internal process of the EXEC, refer to the trace function [Event acquisition and release at data access].

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).  
(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)

An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

An error is not returned even if the virtualization information at the time of switching the CPU operation mode is obtained for a model or virtual machine that does not support the virtualization support function. Since no CPU operation mode switching occurs, no trace data is acquired.

[Precautions when the initial stop / standby mode debug environment is enabled]

(1) While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) The settings related to CPU trace and slave trace are set in the currently selected physical CPU (PE) or slave resource.

(2) Switching of the physical CPU (PE) or slave resource is performed by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL / EX\_UNIT\_SLAVEIDSEL).

[Notes for debugger developers]

(1) If parameters different from the last time are set, the trace data is cleared.

(2) When the trace output type (external trace, internal trace) is changed, the setting value is initialized. Note that the trace output type is changed using ex\_socunitinfo2 (EX\_UNIT\_TRCSEL).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_m\_trcbufcont **[本関数は未サポート]** | **SYNC** | **EMU** | **－** | **BRK** | **－** | **TRC** | **－** | **TIM** | **－** | **COV** | **－** | **HOT** | **－** |
| **ASYNC** | **EMU1** | **－** | **EMU2** | **－** | **BRK1** | **－** | **BRK2** | **－** | **TRC1** | **－** | **TRC2** | **－** |

[Function]

TCUのトレースバッファの設定

[Format]

int ex\_m\_trcbufcont( USHORT func, EXTRCBUF \*trc\_bcont )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <--> | ---------- | -- |
| EXTRCBUF | \*trc\_bcont | <--> | ---------- | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | uncomp\_count | <--> | ---------- | -- |
|  | USHORT | nonbr\_count | <--> | ---------- | -- |
|  | USHORT | bufsuspend | <--> | ---------- | -- |
|  | USHORT | bufunsuspend | <--> | ---------- | -- |
|  | USHORT | bufresume | <--> | ---------- | -- |
| }EXTRCBUF | | |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT 本関数は未サポート

[Description]

[デバッガ関係者向け注意事項]

(1)ex\_rh\_trcbufcontでサポートするため、本関数は、常にEX\_NOSUPRTを返す。

##### ex\_rh\_trcbufcont

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **×** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **○** | **EMU2** | **○** | **BRK1** | **○** | **BRK2** | **○** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

Setting / reference of TCU trace buffer

[Format]

int ex\_rh\_trcbufcont( USHORT func, USHORT trc\_buftype, EXRHTRCBUF \*trc\_bcont )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| USHORT | trc\_buftype | <I > | Trace buffer type | ○ |
| EXRHTRCBUF | \*trc\_bcont | <IO> | Trace buffer status | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | bufnum | < O> | Number of target trace buffers installed | ○ |
|  | USHORT | bufsuspend | <IO> | Instruction / access execution pending request generation point value | ○ |
|  | USHORT | bufunsuspend | <IO> | Release point value of instruction / access execution pending request | ○ |
| }EXRHTRCBUF | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | Setting |
|  | EX\_REF | reference |
|  | | |
| trc\_buftype | | |
|  | EX\_TRCLTB\_CPUTCMBUFF | Specify local trace buffer (TCU\_LTB (for CPU + TCM / LRM)) for PE |
|  | EX\_TRCGTB\_BUFF | Global trace buffer (TCU\_GTB) specification |
|  | EX\_TRCLTB\_SLAVE | 【For RH850 G3 generation MCU】  Specify local trace buffer for slave  【For RH850 G4 generation MCU】  Specification of local trace buffer for outside the cluster |
|  | EX\_TRCLTB\_CLUSTER | Specifying local trace buffer for cluster internal |
|  | | |
| bufnum | | |
|  | <0000h – > | When the local trace buffer for PE (TCU\_LTB (for CPU + TCM / LRM)) is specified |
|  | <0000h – > | When global trace buffer (TCU\_GTB) is specified |
|  | <0000h – > | 【For RH850 G3 generation MCU】  When the local trace buffer for slave is specified  【For RH850 G4 generation MCU】  When the local trace buffer for outside the cluster is specified |
|  | <0000h – > | When the local trace buffer for the cluster is specified |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| bufsuspend | | | |
|  | <0001h – bufnum> | PE用ローカルトレースバッファ(TCU\_LTB(CPU+TCM/LRM用))指定時  (Default:0001h) | |
|  |
|  | <0001h – bufnum> | グローバルトレースバッファ(TCU\_GTB)指定時  (Default:0001h) | |
|  | <0001h – bufnum> | 【RH850 G3世代MCUの場合】  スレーブ用ローカルトレースバッファ指定時(Default:0001h)  【RH850 G4世代MCUの場合】  クラスタ外部用ローカルトレースバッファ指定時(Default:0001h) | |
|  | <0001h – bufnum> | クラスタ内部用ローカルトレースバッファ指定時(Default:0001h) | |
|  | | | |
| bufunsuspend | | | |
|  | <0000h – (bufsuspend-1)> | | PE用ローカルトレースバッファ(TCU\_LTB(CPU+TCM/LRM用))指定時  (Default:0000h) |
|  |  | |  |
|  | <0000h – (bufsuspend-1)> | | グローバルトレースバッファ(TCU\_GTB)指定時  (Default:0000h) |
|  | <0000h – (bufsuspend-1)> | | 【RH850 G3世代MCUの場合】  スレーブ用ローカルトレースバッファ指定時(Default:0000h)  【RH850 G4世代MCUの場合】  クラスタ外部用ローカルトレースバッファ指定時(Default:0000h) |
|  | <0000h – (bufsuspend-1)> | | クラスタ内部用ローカルトレースバッファ指定時(Default:0000h) |
|  | | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_TRACEON Starting trace

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET(0)：トレースバッファタイプで指定したバッファに対するトレースバッファ状態の設定

EX\_REF(1)：トレースバッファタイプで指定したバッファに対するトレースバッファ状態の参照

(2)trc\_buftype

\*trc\_bcontで設定/参照するトレースバッファタイプを指定する。

EX\_TRCLTB\_CPUTCMBUFF(0)：設定/参照するバッファはPE用ローカルトレースバッファ

(TCU\_LTB(CPU+TCM/LRM用))

EX\_TRCLTB\_GRMGIOBUFF(1)：設定/参照するバッファはトレースローカルバッファ

(TCU\_LTB(GRM+GIO用))

EX\_TRCGTB\_BUFF(2) ：設定/参照するバッファはグローバルトレースバッファ(TCU\_GTB)

EX\_TRCLTB\_SLAVE(3) ：設定/参照するバッファはスレーブ用ローカルトレースバッファ/クラスタ内部用ローカルトレースバッファ

EX\_TRCLTB\_CLUSTER(4)　 ：設定/参照するバッファはクラスタ外部用ローカルトレースバッファ

(3)bufnum

実際にデバイス上に搭載されているtrc\_buftypeで指定したバッファの段数を返す。

なお、func=EX\_SET/EX\_REFともにEXECから情報を返す。

(4)bufsuspend

命令実行/データアクセス実行の保留要求を発生させるバッファの段数ポイントを指定する。

bufnumの値より大きい値を設定した場合は、エラー(EX\_PE\_DATA)を返す。

(5)bufunsuspend

命令/アクセス実行の保留要求を解除するバッファの段数ポイントを指定する。

bufunsuspendの値がbufsuspendの値以上の場合は、エラー(EX\_PE\_DATA)を返す。

[動作説明]

ノンリアルタイムトレースモード時の命令実行/データアクセス実行の保留要求/解除を行うTCU内の

ローカルトレースバッファ/グローバルトレースバッファの段数ポイントを設定する。

リアルタイムトレースモード指定時は、本パラメータ設定内容の影響を受けない。

なお、各バッファの設定可能範囲は、func=EX\_REF時のbufnumで参照すること。

(ローカルトレースバッファ、グローバルトレースバッファとは、実際にトレースパケットが格納されるトレースメモリのことではなく、各トレース生成部(TCU\_CPU, TCU\_TCM, TCU\_GRM, TCU\_GIOなど)から出力されるトレースパケットを一時格納するためのバッファである。)

トレース起動中は、設定が行えないためエラー応答する。

トレース出力タイプに内蔵トレースもしくは外部トレースのどちらも選択されていなかった場合は、エラー(EX\_DE\_TRCDISABLE)を返す。

また、デバイスによってトレース機能(TCU)が未実装の場合もあり、この場合はエラーを返す。このエラーを返すときの詳細は、トレース関数の[各I/Fごとの設定対象]を参照。

トレース機能(TCU)の実装有無は、ex\_rh\_getenvにて参照する。

ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。

(次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に設定可能とする)

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[マルチコアおよびスレーブ対応時の注意点]

(1)trc\_buftype=EX\_TRCLTB\_CPUTCMBUFF(0)指定時は、現在選択されている物理CPU(PE)に対して有効となる。

(2)trc\_buftype=EX\_TRCLTB\_CPUTCMBUFF(0)以外の指定では、物理CPU(PE)の選択に関係ない。

(3)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[デバッガ開発者向け注意事項]

(1)命令/アクセス実行保留要求を発生させてから実際に停止するまでにタイムラグがあるため、bufsuspendへの設定値が大き過ぎるとノンリアルタイムトレースモードでもトレース・データのロストが発生する場合がある。

(2)設定を変更してもロストを完全に抑止するわけではないため、適正値を示すことはできない。

(3)前回と異なるパラメータを設定した場合、トレース・データのクリアを行う。

(4)トレース出力タイプ(外部トレース、内蔵トレース)を変更したとき、設定値は初期化される。なお、トレース出力タイプの変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

##### ex\_trcrawmeminfo

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SYNC | EMU | ○ | BRK | ○ | TRC | × | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | × | TRC2 | × |

[Function]

Obtain information necessary for trace raw data read.

[Format]

int ex\_trcrawmeminfo( EXTRCRAWMEMINFO \*info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | 対象 |
| EXTRCRAWMEMINFO | \*info | < O> | Information required for reading raw trace data | ○ |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| typedef struct | |  |  |  |
| { | |  |  |  |
| ULONG | maxsize | < O> | Number of raw data frames stored in trace memory | ○ |
| ULONG | totaltimel | <--> | Unused (0 fixed) | -- |
| ULONG | totaltimeh | <--> | Unused (0 fixed) | -- |
| ULONG | dbg\_mba | < O> | Unused (0 fixed)  Base address of RCU monitor area (debug area) Moved definition to ex\_rh\_getenv. (This parameter is kept for compatibility, but use ex\_rh\_getenv.) | ○ |
| }EXTRCRAWMEMINFO | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| maxsize | | |
|  |  | Number of raw data frames being traced |
|  | | |
| dbg\_mba | | |
|  |  | Base address of RCU monitor area (debug area) |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_TRACEON Starting trace

EX\_DE\_M\_TRCMEMACCFAILED Trace memory access failed

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_FE\_EMUSTORETRACE The emulator failed to record a trace message

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)maxsize

Returns the number of valid frames stored in the trace memory.

(2)dbg\_mba

Returns the base address of the RCU monitor area (debug-only area).  
The returned value is the value of the DBG\_MBA register of the RCU.  
The start address for transition from the user mode to the debug mode during a break operation is a value obtained by adding 0xb0 to this parameter dbg\_mba.  
Some devices have different values for each physical CPU (PE). (It is provisional until the implementation of ex\_rh\_getenv. After implementation of ex\_rh\_getenv, this parameter will be moved there.)

[Description of operation]

Returns the number of valid frames stored in the trace memory.  
If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.  
Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.  
Also, during trace activation, since trace data cannot be obtained by ex\_trcrawmemread, this I / F also returns an error response (EX\_SE\_TRACEON).  
If there is no trace data, maxsize = 0 and no error response.  
  
An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).  
(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".  
  
When the IE850A emulator is connected, trace messages may fail to be recorded between the device and the emulator. In this case, an error (EX\_FE\_EMUSTORETRACE) is returned because it cannot be guaranteed that all the recorded trace messages are correct.

[Precautions when the initial stop / standby mode debug environment is enabled]

(1) While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Notes on multi-core and slave support]

(1) Since the trace memory is commonly used, it is not related to the selection of the physical CPU (PE) and the slave resource. However, since the base address of the monitor area of the RCU may be different for each physical CPU (PE), it is necessary to switch to obtain the information of each PE.

(2) Switching of the physical CPU (PE) is performed by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[Notes for debugger developers]

(1) The value specified by ex\_trcrawmemread must be specified not to exceed the value of maxsize of this function.

(2) The frame at the time of reading the raw trace data is not a unit in which trace data can be analyzed (trace message), but a storage unit in the trace memory, and requires attention.  
The number of frames required for analyzing one trace data varies depending on the trace message, and one to three frames are required. (The number of required frames is determined by checking the trace message head flag and the trace type in the trace packet.)

(3) When the EXEC I / F that changes the setting of the trace function is issued, the trace data is cleared. Therefore, it is necessary to acquire the trace data before changing the setting of the trace function.

##### ex\_trcrawmemread

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SYNC | EMU | ○ | BRK | ○ | TRC | × | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | × | TRC2 | × |

[Function]

Trace raw data read

[Format]

int ex\_trcrawmemread( ULONG trcrawmemaddr, ULONG trcrawmemnum, UCHAR \* trcrawmembuf )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | 対象 |
| ULONG | trcrawmemaddr | <I > | Start address of the trace raw data frame to be read | ○ |
| ULONG | trcrawmemnum | <I > | Number of trace raw data frames to be read | ○ |
| UCHAR \* | trcrawmembuf | < O> | Buffer pointer to return raw trace data | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| trcrawmemaddr | | |
|  |  | Start address of the trace raw data frame to be read |
|  |  |  |
| trcrawmemnum | | |
|  |  | Number of trace raw data frames to be read |
|  |  |  |
| \*trcrawmembuf | | |
|  |  | Buffer pointer to return raw trace data |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_TRACEON Starting trace

EX\_SE\_NOTRACEBLOCK Trace not found

EX\_DE\_M\_TRCMEMACCFAILED Trace memory access failed

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_FE\_EMUSTORETRACE The emulator failed to record a trace message

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)trcrawmemaddr

Specify the start address of the trace raw data frame to be read.  
The address of the trace raw data frame is an address where the oldest data stored in the trace memory is 0.

(2)trcrawmemnum

Specifies the number of trace raw data frames to be read.

(3)trcrawmembuf

Specify the pointer of the buffer to return the raw trace data.  
When TCU trace memory is used, trcrawmembuf stores data of trcrawmemnum × 16 bytes.

[Description of operation]

Trace raw data for trcrawmemnum from the frame specified by trcrawmemaddr is stored in trcrawmembuf. If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.  
Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error (EX\_DE\_TRCNOMOUNT) is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function. In addition, an error response (EX\_SE\_TRACEON) is returned during trace activation because trace data cannot be acquired.  
If there is no trace data, an error response (EX\_SE\_NOTRACEBLOCK) is returned.  
An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).  
(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".  
  
When the IE850A emulator is connected, trace messages may fail to be recorded between the device and the emulator. In this case, an error (EX\_FE\_EMUSTORETRACE) is returned because it cannot be guaranteed that all the recorded trace messages are correct.

[Precautions when the initial stop / standby mode debug environment is enabled]

(1) While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Precautions for multi-core]

(1) Since the trace memory is used in common by all physical CPUs (PEs), it is not related to the selection of the physical CPUs (PEs).

[Notes for debugger developers]

(1) If EX\_NON\_STOP (do not end tracing) is specified in cond.fullstop in ex\_trcmode, overwriting occurs. In this case, care must be taken because the oldest frame may not be a delimited frame where trace data can be analyzed.

(2) The frame at the time of reading the raw trace data is not a unit in which trace data can be analyzed (trace message), but a storage unit in the trace memory, and requires attention.  
The number of frames required for analyzing one trace data varies depending on the trace message, and one to three frames are required. (The number of required frames is determined by checking the trace message head flag and the trace type in the trace packet.)

When using TCU internal trace memory: 16 bytes / frame

Table 3 18 Configuration of One Frame When Using TCU Internal Trace Memory

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Bit [7: 1] of 1st byte to 8th byte | Bit0 of the 8th byte | 9th byte to 16th byte |
| 1 | Reserve data | Trace message start flag (1: Trace message start 0: Not at the beginning of the trace message) | Trace packet |

(3) When the EXEC I / F that changes the setting of the trace function is issued, the trace data is cleared. Therefore, it is necessary to acquire the trace data before changing the setting of the trace function.

(4) When the built-in trace is selected as the trace output type, the format of the trace message read by ex\_trcrawmemread is "G3 generation trace control unit (TCU) function specification" "G4 generation trace control unit (TCU) function See Specifications. Even when the external trace is selected, the format of the trace message is the same as that of the internal trace. However, there are unique messages when using external tracing. See the table below for details. Also, change the trace output type with ex\_socunitinfo2 (EX\_UNIT\_TRCSEL).

Table 3 19 Unique trace message when external trace is selected

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Message name | Message (128-bit notation) | Remarks |
| 1 | Lost off device | 0x0000\_0000\_0000\_0001\_0000\_0000\_FFFF\_FFFF | Issued when a trace message is lost in the POD. \* 1 |
| 2 | Invalid message <For IE850> | 0x0000\_0000\_0000\_0002\_0000\_0000\_0000\_0000 | An invalid message may be issued due to the storage format of the IE850 message. Bit 65 is set as an invalid flag, and all others are set to “0”. |
| 3 | Invalid message <For IE850A> | 0x0000\_0000\_0000\_0003\_0000\_0000\_0000\_0000 | An invalid message may be issued due to the IE850A message storage format. Bit 65 (invalid flag) and bit 64 (head flag) are set to “1”, and all others are set to “0”. |

\* 1 When a “lost outside device” message is received, the trace message immediately preceding that message is invalidated. However, it does not become “invalid message” (bit65 = 1).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_trchwtune **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

トレース・データ取得関連のハードウェア調整機能

[Format]

int ex\_trchwtune( USHORT func, EXTRCHWTUNE \*conf )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <--> | ---------- | -- |
| EXTRCHWTUNE | \*conf | <--> | ---------- | -- |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| typedef struct | |  |  |  |
| { | |  |  |  |
| USHORT | testresults | <--> | ---------- | -- |
| short | idelay | <--> | ---------- | -- |
| ULONG | clockdivision | <--> | ---------- | -- |
| ULONG | datawidth | <--> | ---------- | -- |
| ULONG | dummy | <--> | ---------- | -- |
| }EXTRCHWTUNE | |  |  |  |

[Arguments]

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

##### ex\_rh\_trchwevncopeinfo

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | × |

[Function]

Refers to hardware event number information corresponding to a single event number at trace startup

[Format]

int ex\_rh\_trchwevncopeinfo( USHORT event, USHORT \*num, USHORT \*hw\_event )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | Target |
| USHORT | event | <I > | Event number for acquiring hardware event number | ○ |
| USHORT | \*num | < O> | Number of hardware event numbers | ○ |
| USHORT | \*hwevent | < O> | Hardware event number information | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |
| --- | --- |
| event | |
|  | Single event number for acquiring the hardware event number |
|  | |
| \*num | |
|  | Number of hardware event numbers |
|  | |
| \*hwevent | |
|  | Hardware event number information |
|  | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

[Description]

[Argument description]

(1)event

Specify the single event number for which the hardware event number was obtained.

(2)num

Returns the number of hardware event numbers corresponding to a single event number.

(3)hw\_event

Returns hardware event number information corresponding to a single event number.

[Description of operation]

Returns the hardware event number information corresponding to the specified single event number at the time of the last trace activation.

If the single event number is a range event, a plurality of hardware event numbers will be returned.  
If neither the internal trace nor the external trace is selected as the trace output type, an error (EX\_DE\_TRCDISABLE) is returned.  
Note that the trace function (TCU) may not be implemented depending on the device. In this case, an error is returned. For details on returning this error, refer to [Setting target for each I / F] of the trace function.

ex\_rh\_evncond If a single event number that is not registered is specified, EX\_NOERROR is returned with num = 0.  
If the execution of the EXEC has not been completed, EX\_FE\_COMM is returned with num = 0.  
The trace function (TCU) may not be implemented depending on the device. In that case, EX\_NOSUPRT is returned.

An EX\_SE\_USRPGMRUN error occurs during hot plug-in RUN (from connection to forced break).

(As a future specification after the next development version, it can be set during hot plug-in RUN as well as during normal RUN)

An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return. For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".

[Precautions for debuggers]

(1) When the trace data is acquired under the section trace condition and the branch trace is enabled, a watch point message indicating the start / end of the section (including the information of the corresponding hardware event number) is recorded.

When the event number managed by the debugger and the hardware event number included in the watch point message correspond, the execution address of the section end condition can be grasped. Instruction completion processing can be performed. However, this is limited to the case where an instruction fetch and a point event are set in the section start condition and section end condition.

##### ex\_rh\_trctune

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | × |

[Function]

Trace-related adjustment functions in the trace module (TCU) and emulator (IE850)

[Format]

int ex\_rh\_trctune( USHORT func, USHORT trc\_tunetype, ULONG \*trc\_tunecond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| USHORT | trc\_tunetype | <I > | Adjustment function type | ○ |
| ULONG | \*trc\_tunecond | <IO> | Information for each adjustment function type | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | Setting |
|  | EX\_REF | reference |
|  | | |
| trc\_tunetype | | |
|  | EX\_TRC\_CPUSYNCMSG | Select the output method of the synchronization message generated by TCU\_CPU or TCU for slave |
|  | EX\_TRC\_CPUWATCHMSG | Selection of watch point message type generated by TCU\_CPU or TCU for slave |
|  | EX\_TRC\_CPUMSGCHG | Selection of output method of direct branch message generated by TCU\_CPU |
|  | EX\_TRC\_CPUDBMSG | Selection of message output generated by TCU\_CPU in debug mode |
|  | EX\_TRC\_WPTSMSG | Set the generation function of the watch point timestamp message (WPTSM) generated by TCU\_CPU or TCU for slave |
|  | EX\_TRC\_TIMESYNCMSG | Selecting the generation cycle of the timestamp synchronization message |
|  | EX\_TRC\_GBUFCADET | Setting of global trace buffer continuous access detection function |
|  | EX\_TRC\_GETGTS | Refer to the current value of the global time stamp measurement counter |
|  | EX\_TRC\_TRCRAMOPTION | Setting trace memory options |
|  | EX\_TRC\_TRCOUTIFOPTION | Trace output interface settings |
|  | EX\_TRC\_PODIDELAY | \* Supported only when external trace is enabled |
|  | EX\_TRC\_EMUIDELAY | Setting the skew adjustment circuit of the trace data bus on the POD |
|  | EX\_TRC\_PODCONVMSG | \* Supported only when external trace is enabled |
|  | EX\_TRC\_PODFIFOCOUNT | Setting the skew adjustment circuit of the trace data bus on the emulator (IE850) |
|  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \*trc\_tunecond | | | | | |
|  | Information for each adjustment function type | | | | |
|  | | | | |
|  | <trc\_tunetype = EX\_TRC\_CPUSYNCMSG> | | | |
|  | trc\_tunecond [0] (TRC\_CTRL, TRT\_CTRL, TRG\_CTRL, TRP\_CTRL, TRE\_CTRL (SYM bit) | 256メッセージごとの同期化メッセージ出力の有効/無効 | | |
|  | 0 | | 無効 |
|  | 1 | | 有効(Default) |
|  | trc\_tunecond[1]  (TRC\_CTRL  SYXビット) | 一定数の命令実行(分岐なし)ごとの同期化メッセージ出力の有効/無効  ※TCU\_CPUが対象 | | |
|  | 0 | | 無効 |
|  | 1 | | 有効(Default) |
|  | trc\_tunecond[2]  (TRC\_CRL  SXW[2:0]ビット) | 一定数の命令実行(分岐なし)での同期化メッセージ出力有効時の連続実行検出アドレス範囲  ※trc\_tunecond[1]=1の時のみ有効  ※trc\_tunecond[1]=0の時は、trc\_tunecond[2]=0(固定)を指定すること  ※TCU\_CPUが対象 | | |
|  |  | 0 | | 10bit範囲(+000h～+3FEH)(Default) |
|  |  | 1 | | 9bit範囲(+000h～+1FEH) |
|  |  | 2 | | 8bit範囲(+000h～+0FEH) |
|  |  | 3 | | 7bit範囲(+000h～+07EH) |
|  |  | 4 | | 6bit範囲(+000h～+03EH) |
|  |  | 5 | | 5bit範囲(+000h～+01EH) |
|  |  | 6 | | 4bit範囲(+000h～+00EH) |
|  |  | その他 | | 設定禁止 |
|  | trc\_tunecond[3]  (TRC\_CTRL, TRT\_CTRL, TRG\_CTRL, TRP\_CTRL, TRE\_CTRL  SYI[1:0]ビット) | 同期化要因用端子(MSYNZ)による同期化メッセージ出力有効/無効 | | |
|  | 0 | | 無効 (内蔵トレース選択時はDefault) |
|  | 1 | | MSYNZ端子のH⇒Lのエッジ検出により同期化メッセージ出力(外部トレース選択時はDefault) |
|  | 2 | | MSYNZ端子の入力に依らず常に同期化メッセージ出力  (分岐実行時にのみ直接分岐/間接分岐同期化メッセージを出力) |
|  | 3 | | MSYNZ端子のLレベル検出により同期化メッセージ出力 |
|  | その他 | | 設定禁止 |
|  | | | | |
|  | <trc\_tunetype=EX\_TRC\_CPUWATCHMSG>時 | | | |
|  | trc\_tunecond[0]  (TRC\_CTRL, TRT\_CTRL, TRG\_CTRL, TRP\_CTRL, TRE\_CTRL  WMTビット) | ウォッチ・ポイントメッセージタイプの選択 | | |
|  | 0 | 拡張なしウォッチ・ポイントメッセージ | |
|  | 1 | 拡張ありウォッチ・ポイントメッセージ(Default)  ※各イベントチャネルの検出結果を付加 | |
|  | その他 | 設定禁止 | |
|  | | | | |
|  | <trc\_tunetype=EX\_TRC\_CPUMSGCHG>時 | | | |
|  | trc\_tunecond[0]  (TRC\_TFE00  PDBnビット) | 直接分岐メッセージタイプの選択  ※TCU\_CPUが対象 | | |
|  | 0 | 直接分岐メッセージ(Default) | |
|  | 1 | 間接分岐メッセージに変換  (パケット数が直接分岐時の2倍になる) | |
|  | その他 | 設定禁止 | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_CPUDBMSG>時 | | | |
|  | trc\_tunecond[0]  (TRC\_CTRL  DTEビット) | デバッグモード時のメッセージ出力有効/無効  ※TCU\_CPUが対象 | | |
|  | 0 | デバッグモード時は無効(Default) | |
|  | 1 | デバッグモード時も有効 | |
|  | その他 | 設定禁止 | |
|  | | | | |
|  | <trc\_tunetype= EX\_TRC\_WPTSMSG>時 ※ RH850 G4世代MCU専用 | | | |
|  | trc\_tunecond[0]  (TR\*\_CTRL  WPTビット) | ウォッチポイントタイムスタンプメッセージ(WPTSM)の生成の有効／無効の設定／参照 | | |
| 0 | 無効 (Default) | |
| 1 | 有効 | |
| その他 | 設定禁止 | |
|  |  | | |
|  | | | | |
|  | <trc\_tunetype=EX\_TRC\_TIMESYNCMSG>時 | | | |
|  | trc\_tunecond[0]  (TRB\_TSC  SYTEビット) | タイムスタンプ同期化メッセージの発生周期の有効/無効 | | |
|  | 0 | 無効 | |
|  | 1 | 有効(Default) | |
|  | その他 | 設定禁止 | |
|  | trc\_tunecond[1]  (TRB\_TSC  SYTC[27:4]ビット) | タイムスタンプ同期化メッセージの発生周期  ※trc\_tunecond[0]=1の時のみ有効  ※trc\_tunecond[0]=0の時は、trc\_tuncond[1]=00000000h(固定)を指定すること | | |
|  | <00000000h－00FFFFFF> | | 15クロック～256Mクロック  ※SYTC[3:0]=Fh固定となっているので設定変更できない。 |
|  | その他 | | 設定禁止 |
|  | | | | |
|  | <trc\_tunetype=EX\_TRC\_GBUFCADET>時 ※ RH850 G4世代MCU専用 | | | |
|  | trc\_tunecond[0]  (TRB\_CAEN-  CAEN) | 連続メッセージ生成部(TCU\_CON\_GEN)での連続アクセス検出機能の有効／無効 | | |
| 0 | 無効 (Default) | |
| 1 | 有効 | |
| その他 | 設定禁止 | |
|  | | |
|  | trc\_tunecond[1]  (TRB\_CAEN-  CMAX) | 連続アクセスとして検出するアクセス間隔の最大期間(最大クロック数) | | |
| 0x0-0xF | アクセス間隔の最大期間 (Default値：0xE ※1) | |
| その他 | 設定禁止 | |
| ※1 本機能未対応の場合のDefault値は0 | | |
|  | trc\_tunecond[2]  (TRB\_SCS[31:0]) | メッセージソースクライアントn(トレース情報検出部n)(＠SRC4～0＝n)を連続メッセージ生成部(TCU\_CON\_GEN)での連続アクセス検出機能の検出対象 (ビット指定) | | |
| 0b | クライアントを選択しない (Default) | |
| 1b | クライアントを選択する | |
|  | | |
|  | | | | |
|  | <trc\_tunetype=EX\_TRC\_GETGTS>時 ※ RH850 G4世代MCU専用 | | | |
|  | trc\_tunecond[0]  (TRB\_TSMP) | グローバルタイムスタンプ(絶対時間)計測カウンタ(32 ビット)の現在値 (参照のみ) | | |
|  | <00000000h－FFFFFFFFh> | | |
|  |  | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_TRCRAMOPTION>時 | | | |
|  | trc\_tunecond[0]  (内蔵トレース選択時：TRM\_CTRL  TAIビット) | トレースメモリの記録単位選択 | | |
|  | 0 | 1パケット=128bit(メッセージ先頭フラグ格納) (Default) | |
|  | 1 | 1パケット=64bit(メッセージ先頭フラグなし) | |
|  | その他 | 設定禁止 | |
|  | 外部トレース選択時は0固定とし、設定値は無視する。また、参照値は0固定となる。 | | |
|  | trc\_tunecond[1]  (内蔵トレース選択時：TRM\_ADDR  外部トレース選択時(IE850)：IFPGA  TRCADRCNT)  外部トレース選択時(IE850A)：FPGA  TRC\_MEM\_FULBRK\_ADDR) | フルブレーク時のフルブレーク発生後に出力されるトレースメッセージの記録余裕フレームの設定 | | |
|  | <00000000h－> (単位：フレーム(パケット)) | | 内蔵トレースもしくは外部トレース(IE850)選択時  トレースメモリの先頭アドレス位置～(トレースメモリ最大サイズ－n)  ※nは、trc\_tunecond[0]=0時：2, trc\_tunecond[0]=1時：1  外部トレース(IE850A)選択時  トレースメモリの最大フレームから本パラメータの設定値を減じた位置でフルブレークイベントを発生させる。  ただし、ex\_setenv - trcmem\_sizeの設定値によっては、本パラメータは無視される。 |
|  | ※フルブレーク以外は、このパラメータは無効。 | | |
|  | trc\_tunecond[2]  (内蔵トレース選択時：TRM\_SIZE) | 内蔵トレースメモリの使用サイズ選択 | | |
|  | <00000000h－> (単位：バイト) | | 0～(トレースメモリ最大サイズ)  ※デバイス、およびtrc\_tunecond[0]により、設定可能な値が異なる。 |
|  | 外部トレース選択時は、設定値は無視する。参照値は不定値となる。 | | |
|  | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_TRCOUTIFOPTION>時 | | | | | | |
|  | trc\_tunecond[0]  (TRO\_CTRL  MTPビット) | メッセージの多重出力の設定 | | | | | |
|  | 0 | 多重出力しない(1重出力) | | | | |
|  | 1 | 多重出力する(2重出力) | | | | |
|  | その他 | 設定禁止 | | | | |
|  |  | | | | | |
|  | trc\_tunecond[1]  (TRO\_CTRL  MSEビット) | メッセージ・スタート/エンド出力信号の有効端子数の設定 | | | | | |
|  | 0 | 1本 (設定禁止) | | | | |
|  | 1 | 2本 | | | | |
|  | その他 | 設定禁止 | | | | |
|  |  | | | | | |
|  | trc\_tunecond[2]  (TRO\_CTRL  MDW[2:0]ビット) | メッセージ・データ出力信号の有効な出力幅を設定 | | | | | |
|  | trc\_tunecond[0] | | =0 | | =1 | |
|  | 0 | | 2bit | | 設定禁止 | |
|  | 1 | | 4bit | | 2bitx2重 | |
|  | 2 | | 8bit | | 4bitx2重 | |
|  | 3 | | 16bit | | 8bitx2重 | |
|  | 4 | | 24bit | | 12bitx2重 | |
|  | 5 | | 32bit | | 16bitx2重 | |
|  | 6 | | 48bit | | 24bitx2重 | |
|  | 7 | | 64bit | | 32bitx2重 | |
|  | その他 | | 設定禁止 | | 設定禁止 | |
|  |  | | | | | |
|  | trc\_tunecond[3]  (TRO\_CTRL  MOEビット) | メッセージ・スタート/エンド出力信号、メッセージ・データ出力信号の同期エッジの設定 | | | | | |
|  | 0 | メッセージ・クロック出力信号の立ち上がりエッジ(SDR) | | | | |
|  | 1 | メッセージ・クロック出力信号の両エッジ(DDR) | | | | |
|  | その他 | 設定禁止 | | | | |
|  |  | | | | | |
|  | trc\_tunecond[4]  (TRO\_CTRL  MCD[2:0]ビット) | メッセージ・クロック出力信号のシステム・クロック(CPUクロック)に対する分周比を設定 | | | | | |
|  | trc\_tunecond[0]= | | 0 | | 1 | |
|  | trc\_tunecond[3]= | | 0 | 1 | 0 | 1 |
|  | 0 | | 1分周 | 設定禁止 | 設定禁止 | 設定禁止 |
|  | 1 | | 2分周 | 2分周 | 2分周 | 設定禁止 |
|  | 2 | | 設定禁止 | 設定禁止 | 設定禁止 | 設定禁止 |
|  | 3 | | 4分周 | 4分周 | 4分周 | 4分周 |
|  | 4 | | 6分周 | 6分周 | 設定禁止 | 設定禁止 |
|  | 5 | | 8分周 | 8分周 | 8分周 | 8分周 |
|  | 6 | | 設定禁止 | 設定禁止 | 設定禁止 | 設定禁止 |
|  | 7 | | 16分周 | 16分周 | 16分周 | 16分周 |
|  | その他 | | 設定禁止 | 設定禁止 | 設定禁止 | 設定禁止 |
|  |  | | | | | |
|  | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_PODIDELAY>時 | | |
|  | trc\_tunecond[0]  (TFPGA-  IDELAY\_RSTZ  IDELAY\_RSTZビット) | トレース・データバスのスキュー調整回路のリセット | |
|  | 0 | リセット |
|  | 1 | リセット無効 (Default) |
|  | その他 | 設定禁止 |
|  |  | |
|  | trc\_tunecond[1]  (TFPGA-  IDELAY\_MCKO\_SET  MCKO\_INCビット) | メッセージ・クロック出力端子の遅延TAPをtrc\_tunecond[3]で設定した値までインクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | インクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[2]と同時に1を設定しないこと) | |
|  | trc\_tunecond[2]  (TFPGA-  IDELAY\_MCKO\_SET  MCKO\_DECビット) | メッセージ・クロック出力端子の遅延TAPをtrc\_tunecond[3]で設定した値までデクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | デクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[1]と同時に1を設定しないこと) | |
|  | trc\_tunecond[3]  (TFPGA-  IDELAY\_MCKO\_SET  MCKO\_TAP[7:0]ビット) | メッセージ・クロック出力端子の遅延TAP値を設定 | |
|  | 0～255 | TAP値 |
|  | その他 | 設定禁止 |
|  |  | |
|  | trc\_tunecond[4]  (TFPGA-  IDELAY\_MDO\_SET  MDO\_INCビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAPをtrc\_tunecond[6]で設定した値までインクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | インクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[5]と同時に1を設定しないこと) | |
|  | trc\_tunecond[5]  (TFPGA-  IDELAY\_MDO\_SET  MDO\_DECビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAPをtrc\_tunecond[6]で設定した値までデクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | デクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[4]と同時に1を設定しないこと) | |
|  | trc\_tunecond[6]  (TFPGA-  IDELAY\_MDO\_SET  MDO\_TAP[7:0]ビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAP値を設定 | |
|  | 0～255 | TAP値 |
|  | その他 | 設定禁止 |
|  |  | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_EMUIDELAY>時 | | |
|  | trc\_tunecond[0] | N/A | |
|  | 0を設定する。0以外を設定してもEXECはエラーを返さず無視する。  (EX\_TRC\_PODIDELAYとパラメータ構造を合わせるためのパラメータ) | |
|  | trc\_tunecond[1]  (IFPGA-  IDELAY\_MCKO\_SET  MCKO\_INCビット) | メッセージ・クロック出力端子の遅延TAPをtrc\_tunecond[3]で設定した値までインクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | インクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[2]と同時に1を設定しないこと) | |
|  | trc\_tunecond[2]  (IFPGA-  IDELAY\_MCKO\_SET  MCKO\_DECビット) | メッセージ・クロック出力端子の遅延TAPをtrc\_tunecond[3]で設定した値までデクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | デクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[1]と同時に1を設定しないこと) | |
|  | trc\_tunecond[3]  (IFPGA-  IDELAY\_MCKO\_SET  MCKO\_TAP[5:0]ビット) | メッセージ・クロック出力端子の遅延TAP値を設定 | |
|  | 0～63 | TAP値 |
|  | その他 | 設定禁止 |
|  |  | |
|  | trc\_tunecond[4]  (IFPGA-  IDELAY\_MDO\_SET  MDO\_INCビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAPをtrc\_tunecond[6]で設定した値までインクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | インクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[5]と同時に1を設定しないこと) | |
|  | trc\_tunecond[5]  (IFPGA-  IDELAY\_MDO\_SET  MDO\_DECビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAPをtrc\_tunecond[6]で設定した値までデクリメントする設定 | |
|  | 0 | 無効 |
|  | 1 | デクリメント |
|  | その他 | 設定禁止 |
|  | (trc\_tunecond[4]と同時に1を設定しないこと) | |
|  | trc\_tunecond[6]  (IFPGA-  IDELAY\_MDO\_SET  MDO\_TAP[5:0]ビット) | メッセージ・データ出力端子、メッセージ・スタート/エンド出力端子の遅延TAP値を設定 | |
|  | 0～63 | TAP値 |
|  | その他 | 設定禁止 |
|  |  | |
|  | | | |
|  | <trc\_tunetype=EX\_TRC\_PODCONVMSG>時 | | |
|  | trc\_tunecond[0]  (TFPGA-  TRCCONVSET  BRCONVSET[0]ビット) | トレースメッセージの変換回路の設定 | |
| 0 | 変換しない (Default) |
| 1 | 変換する (本開発バージョンでは設定禁止) |
| その他 | 設定禁止 |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | <trc\_tunetype=EX\_TRC\_PODFIFOCOUNT>時 | | |
|  | trc\_tunecond[0]  (TFPGA-  SFIFO\_CONEN) | カウント動作の有効／無効 | |
| 0 | 無効 |
| 1 | 有効 |
| その他 | 設定禁止 |
|  | |
|  | trc\_tunecond[1]  (TFPGA-  SFIFO\_CMPL/  SFIFO\_CMPH) | コンペア値 | |
|  | <00000000h－FFFFFFFFh> | |
|  |  | |
|  | trc\_tunecond[2]  (TFPGA-  SFIFO\_CONL/  SFIFO\_CONH) | カウント値 (参照のみ可) | |
|  | <00000000h－FFFFFFFFh> | |
|  |  | |
|  | | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_TRACEON Starting trace

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_DE\_TRCDISABLE Trace output is invalid

EX\_PE\_SELTRC Cannot be set for the selected trace resource

EX\_PE\_EXCLUSION 設定値が排他でない

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_DE\_NOSUPRTPODFIFOCNT PODのトレース用FIFOの書き込みカウント機能は未サポート

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET(0)：trc\_tunetypeで指定した調整機能に対する情報の設定

EX\_REF(1)：trc\_tunetypeで指定した調整機能に対する情報の設定状態の参照

(2)trc\_tunetype

trc\_tunecondで指定する調整機能タイプを指定する。

(3)trc\_tunecond

trc\_tunetypeで指定した調整タイプに対応する情報を指定する。

[Description of operation]

トレース機能の各種レジスタの参照/設定を行う。

トレース出力タイプに内蔵トレースもしくは外部トレースのどちらも選択されていなかった場合は、エラー(EX\_DE\_TRCDISABLE)を返す。

なお、デバイスによってはトレース機能(TCU)が未実装の場合もあり、この場合はエラーを返す。このエラーを返すときの詳細は、トレース関数の[各I/Fごとの設定対象]を参照。

<trc\_tunetype=EX\_TRC\_CPUSYNCMSG>時

TCU\_CPUまたはスレーブ用TCUが生成するメッセージに対して同期化メッセージ出力の有効/無効などの情報設定を行う。

なお、同期化メッセージは、本設定に関わらず同期化要因が発生した時点で出力される。(同期化要因については、G3世代TCU機能仕様書を参照のこと。)

また、現在選択されている物理CPU(PE)または、スレーブ資源に設定対象のレジスタがない場合は、エラー(EX\_PE\_SELTRC)を返す。

trc\_tunecond[0]には、256メッセージ生成されるごとに同期化メッセージを出力するかどうかを指定する。

この設定は、あくまで生成されたすべてのメッセージ数が256以上になった場合の同期化メッセージを出力するかどうかを指定する。

trc\_tunecond[1]、trc\_tunecond[2]には、分岐命令実行後、一定数の命令実行の間に次の分岐命令が実行されなかった場合に同期化メッセージを出力するか、また出力する場合の命令実行アドレスの検出範囲を指定する。

trc\_tunecond[3]には、同期化要因端子(MSYNZ)の状態によって同期化メッセージを出力するかを指定する。

<trc\_tunetype=EX\_TRC\_CPUWATCHMSG>時

セクション・トレースやクオリファイ・トレースなどのイベントを使用した条件トレースを行っていた場合、イベント発生時に生成されるウォッチ・ポイントメッセージタイプを指定する。

trc\_tunecond[0]=0の場合、生成されるウォッチ・ポイントメッセージには、サブ・スイッチ情報の有効/無効および各クオリファイ/セクションサブスイッチの状態(ON/OFF)の情報のみ付加される。

trc\_tunecond[0]=1の場合、生成されるウォッチ・ポイントメッセージには、trc\_tunecond[0]=0時の情報に加えて、すべてのハードウェアイベントチャネルごとの検出結果も付加される。

<trc\_tunetype=EX\_TRC\_CPUMSGCHG>時

直接分岐命令実行時に生成される直接分岐メッセージタイプを指定する。

0指定の場合、直接分岐メッセージは変化しないでそのまま直接分岐メッセージとして生成される。この場合、直接分岐メッセージには、分岐先アドレス情報が付加されないため、分岐元アドレスに配置された命令コードを解析して分岐先アドレスを取得する必要がある。

1指定の場合、直接分岐メッセージは間接分岐メッセージとして生成される。間接分岐メッセージでは分岐先アドレス情報も記録されるため、分岐元アドレスに配置された命令コードの解析は必要ないが、メッセージサイズが直接分岐メッセージ時よりも2倍(消費するトレースメモリサイズも2倍)となる。

<trc\_tunetype=EX\_TRC\_CPUDBMSG>時

デバッグモード時にトレース起動を有効にするかどうかを指定する。

0指定の場合、デバッグモード時はトレース停止状態となりメッセージは生成されない。

1指定の場合、デバッグモード時もトレース起動状態は継続されメッセージも生成される。

なお、この設定に関わらずデバッグモードからの復帰命令のメッセージは常に生成される。

<trc\_tunetype=EX\_TRC\_WPTSMSG>時

ウォッチポイントタイムスタンプメッセージ(WPTSM)の生成するかどうかを指定する。

0指定の場合は、TSM とWPM を統合しない。

1指定の場合は、TSM とWPM を統合する。

設定対象は、全ての取得資源(CPUとスレーブ)となる。

設定禁止の値を設定した場合はエラー(EX\_PE\_DATA)を返す。

また、WPTSMの生成機能に非対応のトレース資源は0(無効)固定となる。もし1(有効)を設定した場合は、エラー(EX\_PE\_SELTRC)を返す。

トレース資源がWPTSMの生成機能のサポート状況は、ex\_rh\_getenv – funcinfo[16]-[18]を参照。

なお、RH850 G3世代MCUの場合は、本機能に非対応のため必ずエラー応答(EX\_PE\_DATA)を返す。

<trc\_tunetype=EX\_TRC\_TIMESYNCMSG>時

絶対時間に復元するために生成されるタイムスタンプ同期化メッセージの発生の有効/無効および有効時の発生周期を指定する。

なお、タイムスタンプの同期化要因が発生した場合は、本設定に依らずタイムスタンプ同期化メッセージが生成される。(同期化要因については、G3世代TCU機能仕様書を参照のこと。)

trc\_tunecond[0]には、trc\_tunecond[1]に指定した発生周期のタイミングでタイムスタンプ同期化メッセージ出力を有効にするかどうかを指定する。

trc\_tunecond[1]には、trc\_tunecond[0]=1を指定時に、タイムスタンプ同期化メッセージを出力する発生周期を指定する。

<trc\_tunetype=EX\_TRC\_GBUFCADET>時

trc\_tunecond[0]には、グローバルトレースバッファ連続アクセス検出機能の有効／無効を設定する。

連続メッセージ生成部での連続アクセス検出機能に対応していない場合に有効を設定した場合は、エラー(EX\_PE\_SELTRC)を返す。

trc\_tunecond[1]には、連続アクセスとして検出するアクセス間隔の最大期間(最大クロック数)を設定する。

設定禁止の値を設定した場合はエラー(EX\_PE\_DATA)を返す。

なお、設定可能な値であっても、デバイスで設定禁止の値が存在する。

(デバイスの設定禁止の値：0x0、0x1、0xF)

この場合は、設定してもエラーを返さない。(将来のデバイスで設定可能となる可能性があるため。)

trc\_tunecond[2]には、メッセージソースクライアントn(トレース情報検出部n)(＠SRC4～0＝n)を連続アクセス検出機能の検出対象としてビット単位選択する。(n=31～0)

なお、全てのビットが選択可能ではなく、デバイスで選択禁止されているビットが存在する。

(デバイスの設定禁止のビット：b20(DMA0), b21(DMA1), b22(DTS)以外)

この場合は、設定してもエラーを返さない。(将来のデバイスで設定可能となる可能性があるため。)

グローバルトレースバッファ連続アクセス検出機能のサポート状況は、ex\_rh\_getenv – funcinfo[0].b7を参照。

なお、RH850 G3世代MCUの場合は、本機能に非対応のため必ずエラー応答(EX\_PE\_DATA)を返す。

<trc\_tunetype=EX\_TRC\_GETGTS>時

グローバルタイムスタンプ(絶対時間)計測カウンタ(32 ビット)値をtrc\_tunecond[0]に返す。

参照のみ可能のため、func=EX\_SETで呼び出した場合は、エラー(EX\_PE\_DATA)を返す。

また、絶対時間計測カウンタ値の読み出しに対応していない場合は、エラー(EX\_PE\_SELTRC)を返す。

グローバルタイムスタンプ(絶対時間)計測カウンタの取得のサポート状況は、ex\_rh\_getenv – funcinfo[0].b8を参照。

なお、RH850 G3世代MCUの場合は、本機能に非対応のため必ずエラー応答(EX\_PE\_DATA)を返す。

<trc\_tunetype=EX\_TRC\_TRCRAMOPTION>時

トレースメモリ関連の各設定情報を指定する。

トレース出力タイプが内蔵トレースを選択中は内蔵トレースメモリに対して設定を行う。外部トレースを選択中はエミュレータ(IE850)上などの外部トレースメモリに対して設定を行う。

なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[補足情報]

トレースメモリには、解析単位であるメッセージ単位で記録されるのではなく、トレース情報の生成単位であるパケット単位で記録される。

1パケットは64bitであり、フルストップモードやフルブレークモードでトレース起動している場合、オーバーフローが発生しない状態であれば、パケットだけをトレースメモリに記録しても、記録開始位置からパケットを解析することで、解析単位であるメッセージに変換できる。

しかし、上書きモード(リングモード)でトレース起動している場合や、オーバーフローなどが発生した場合、パケットだけをトレースメモリに記録すると、解析単位であるメッセージの先頭がどのパケットであるか判断できない。

trc\_tunecond[0]には、トレースメモリへの記録単位を指定する。

0指定の場合、1パケットを128bit単位で記録する。(記録される情報には、パケット(64bit)だけでなくメッセージ先頭を示すフラグ(1bit)が付加される)

1指定の場合、1パケットを64bit単位で記録する。(記録する情報はパケットのみ)

なお、内蔵トレース選択中のみ1に設定可能で、外部トレース選択中は0固定として、0以外を設定した場合は無視する。

trc\_tunecond[1]には、トレースメモリのパケット記録開始アドレス位置を指定する。

値はフレーム(パケット)単位で指定する。

trc\_tunecond[0]の値により、指定範囲が異なる。

・trc\_tunecond[0]=0の場合、00000000hから000007FFh(PFC1A時)

・trc\_tunecond[0]=1の場合、00000000hから00000FFFh(PFC1A時)

trc\_tunecond[2]には、トレースメモリの使用サイズを指定する。

指定サイズはバイト単位で指定する。

trc\_tunecond[0]=0の場合は指定範囲を16の倍数バイト、trc\_tunecond[0]=1の場合は指定範囲を8の倍数バイトに指定すること。

なお、内蔵トレース選択時のみ設定は有効で、外部トレース選択時には値を無視する。また、参照した場合は不定値となる。(外部トレース選択時は、同様の設定についてex\_setenvで行うことが可能。)

内蔵トレース選択時の設定可能な最大値は、ex\_rh\_getenv - mcuinfo[0]で取得可能。また、デフォルト値もその値となる。

<trc\_tunetype=EX\_TRC\_TRCOUTIFOPTION>時

トレース出力インターフェース関連の各設定情報を指定する。

外部トレースを選択中のときだけ有効。内蔵トレースを選択中に設定した場合は、エラー(EX\_PE\_SELTRC)を返す。なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

trc\_tunecond[0]には、メッセージ・データの多重出力の有効/無効を設定する。

trc\_tunecond[1]には、メッセージ・スタート/エンド出力信号の有効な端子数を設定する。1固定に設定すること。

trc\_tunecond[2]には、メッセージ・データ出力信号の有効な出力幅を設定する。trc\_tunecond[0]の設定値によって設定可能な出力幅が変化する。

trc\_tunecond[3]には、メッセージ・スタート/エンド出力信号およびメッセージ・データ出力信号の同期エッジを選択する。

trc\_tunecond[4]には、メッセージ・クロック出力信号に対する分周比を設定する。trc\_tunecond[0] およびtrc\_tunecond[3]の設定値によって設定可能な分周比が変化する。

デフォルト値はENVファイルからリードされた値となる。

<trc\_tunetype=EX\_TRC\_PODIDELAY>時

POD上のトレース・データバスのスキュー調整回路の各設定情報を指定する。

外部トレースを選択中のときだけ有効。内蔵トレースを選択中に設定した場合は、エラー(EX\_PE\_SELTRC)を返す。なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

trc\_tunecond[0]には、POD上のトレース・データバスのスキュー調整回路のリセット状態を指定する。

trc\_tunecond[3]には、メッセージ・クロック出力端子の遅延TAP値を設定する。遅延TAP値をインクリメントするかデクリメントするかの設定は、trc\_tunecond[1]～[2]で排他に指定する。両方1もしくは両方0に設定した場合は、エラー(EX\_PE\_EXCLUSION)を返す。

trc\_tunecond[6]には、メッセージ・データ出力端子とメッセージ・スタート/エンド出力端子の遅延TAP値を設定する。遅延TAP値をインクリメントするかデクリメントするかの設定は、trc\_tunecond[4]～[5]で排他に指定する。両方1もしくは両方0設定した場合は、エラー(EX\_PE\_EXCLUSION)を返す。

trc\_tunecond[0]以外は、デフォルト値はENVファイルからリードされた値となる。

<trc\_tunetype=EX\_TRC\_EX\_TRC\_EMUIDELAY>時

エミュレータ(IE850)上のトレース・データバスのスキュー調整回路の各設定情報を指定する。

外部トレースを選択中のときだけ有効。内蔵トレースを選択中に設定した場合は、エラー(EX\_PE\_SELTRC)を返す。なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

trc\_tunecond[0]には、必ず0を設定する。もし、0以外を設定してもEXECはエラーを返さず無視する。

trc\_tunecond[3]には、メッセージ・クロック出力端子の遅延TAP値を設定する。遅延TAP値をインクリメントするかデクリメントするかの設定は、trc\_tunecond[1]～[2]で排他に指定する。両方1もしくは両方0に設定した場合は、エラー(EX\_PE\_EXCLUSION)を返す。

trc\_tunecond[6]には、メッセージ・データ出力端子とメッセージ・スタート/エンド出力端子の遅延TAP値を設定する。遅延TAP値をインクリメントするかデクリメントするかの設定は、trc\_tunecond[4]～[5]で排他に指定する。両方1もしくは両方0に設定した場合は、エラー(EX\_PE\_EXCLUSION)を返す。

デフォルト値はENVファイルからリードされた値となる。

<trc\_tunetype=EX\_TRC\_PODCONVMSG>時

POD上のトレースメッセージの変換回路の設定を行う。

変換回路の実装有無については、EXECでは判定しない。変換回路が実装されていないPODに対して「変換する」(trc\_tunecond[0]=1)に設定したときの動作は保証しない。変換回路の実装有無については、各品種のTFPGAの仕様書を参照。

また、「変換する」に設定するときは、本I/Fによって必ず下記の設定にしなければならない。

* trc\_tunetype=EX\_TRC\_CPUMSGCHG – trc\_tunecond[0]=1h (TRC\_TFE00のPDBnの設定)

<trc\_tunetype=EX\_TRC\_PODFIFOCOUNT>時

PODのトレース用FIFO(SFIFO)の書き込みカウント機能の設定／参照を行う。

「有効」(trc\_tunecond[0]=1)、かつコンペア値>0に設定した時、SFIFOに書き込まれたメッセージ数をカウントし、コンペア一致した時に同期化要求をデバイスに出力する。なお、トレースメッセージのサイズによっては、1メッセージに対して本機能のカウンタは複数カウントされる場合がある。

カウント値(trc\_tunecoud[2])は参照のみ。設定値は無視される。

本機能は、デバイスにより実装されていないものがある。ex\_rh\_getenv – funcinfo[0]により実装有無の情報を取得できる。未実装のデバイスに対して、「有効」を設定した場合は、エラー(EX\_DE\_NOSUPRTPODFIFOCNT)を返す。

外部トレースを選択中のときだけ有効。内蔵トレースを選択中に設定した場合は、エラー(EX\_PE\_SELTRC)を返す。なお、トレース出力タイプ(外部トレース、内蔵トレース)の変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

デフォルト値はENVファイルからリードされた値となる。

[マルチコアおよびスレーブ対応時の注意点]

(1)trc\_tunetypeがEX\_TRC\_CPUSYNCMSG、EX\_TRC\_CPUWATCHMSGの場合、現在選択されている物理CPU(PE)または、スレーブ資源に対して設定する。

(2)trc\_tunetypeがEX\_TRC\_CPUMSGCHG、EX\_TRC\_CPUDBMSGの場合、現在選択されている物理CPU(PE)に対して設定する。

(3)物理CPU(PE) および、スレーブ資源の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL/EX\_UNIT\_SLAVEIDSEL)で行う。

[デバッガ関係者向け注意事項]

(1)パラメータを設定した場合、トレース・データのクリアを行う。

(2)トレース出力タイプ(外部トレース、内蔵トレース)を変更したとき、設定値は初期化される。なお、トレース出力タイプの変更は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

(3)ex\_rh\_trctuneはデバッグ用途のI/Fのため、下記のtrc\_tunetype以外は動作保証の対象外とする。

EX\_TRC\_CPUSYNCMSG, EX\_TRC\_CPUMSGCHG, EX\_TRC\_TIMESYNCMSG, EX\_TRC\_PODCONVMSG, EX\_TRC\_PODFIFCOUNT

(4)トレース出力タイプと接続されているエミュレータ種別によって呼び出し可能なtrc\_tunetypeは異なる。呼び出し可否を下表に示す。

表 3‑20 trc\_tunetypeの呼び出し可否

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| trc\_tunetype | トレース出力先 | 内蔵トレース | 外部トレース | | 備考 |
| エミュレータ種別 | 全て | IE850 | IE850A |
| EX\_TRC\_CPUSYNCMSG | | ○ | ○ | ○ |  |
| EX\_TRC\_CPUWATCHMSG | | ○ | ○ | ○ |  |
| EX\_TRC\_CPUMSGCHG | | ○ | ○ | ○ |  |
| EX\_TRC\_CPUDBMSG | | ○ | ○ | ○ |  |
| EX\_TRC\_WPTSMSG | | ○ | ○ | ○ |  |
| EX\_TRC\_TIMESYNCMSG | | ○ | ○ | ○ |  |
| EX\_TRC\_GBUFCADET | | ○ | ○ | ○ |  |
| EX\_TRC\_GETGTS | | ○ | ○ | ○ |  |
| EX\_TRC\_TRCRAMOPTION | | ○ | ○ | ○ |  |
| EX\_TRC\_TRCOUTIFOPTION | | × | ○ | × |  |
| EX\_TRC\_PODIDELAY | | × | ○ | × |  |
| EX\_TRC\_EMUIDELAY | | × | ○ | × |  |
| EX\_TRC\_PODCONVMSG | | × | ○ | × |  |
| EX\_TRC\_PODFIFOCOUNT | | × | ○ | × |  |

○：正常応答、×：エラー応答(EX\_PE\_SELTRC)

##### ex\_rh\_trcfilter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

Set / reference the trace filter condition.

[Format]

int ex\_rh\_trcfilter( USHORT func, ULONG filter\_type, ULONG \*num, ULONG \*filter\_info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | Argument identifier | Attributes | Explanation | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| ULONG | filter\_type | <I > | Filter type | ○ |
| ULONG | \*num | <IO> | Number of information by filter | ○ |
| ULONG | \*filter\_info | <IO> | Information by filter | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| func | | | | |
|  | EX\_SET | Setting | | |
|  | EX\_REF | reference | | |
| filter\_type | | | | |
|  | EX\_FILTERTYPE\_SPID | SPID filtering setting / reference | | |
| \*num | | | | |
|  | <00000000h-FFFFFFFFh> | | Set the number of filter\_info. \* Returns the number of returned filter\_info when referencing. It is assumed that there is enough filter\_info area for the specified filter type. In addition, \* num should be set and called for all filter types even when referring. | |
| \*filter\_info | | | | |
|  | <EX\_FILTERTYPE\_SPID> (num=1) | | | |
|  | filter\_info[0] | bit0 | | Trace output of SPID0  0：Invalid  1：Enabled (Default) |
|  |  | bit1 | | Trace output of SPID1  0：Invalid  1：Enabled (Default) |
|  |  | ・  ・  ・ | |  |
|  |  | bit31 | | Trace output of SPID31  0：Invalid  1：Enabled (Default) |
|  |  |  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_TRCNOMOUNT Trace function is not implemented

EX\_DE\_SELTRCNOMOUNT The selected trace function is not implemented

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_SE\_ASYNCMODE\_SELPEDSB Unavailable due to non debug target core in asynchronous debug mode

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

Common for each filter\_type

[Argument description]

(1)func

Specify the operation mode.

EX\_SET：Set the value specified for each parameter.

EX\_REF：Returns the current settings to each parameter.

(2)num

Set the value determined by each filter\_type. Also, secure an area for filter\_info [] with the set value.

＜filter\_type=EX\_FILTERTYPE\_SPID＞

[Argument description]

(1)filter\_info[0]

Set / reference the SPID to be filtered in bit units.

[Description of operation]

【G3 generation】

　There is no function entry itself.

【G4 generation】

Set / reference the SPID to be traced by setting the SPID in bit units.  
If filtering by a specific SPID is not performed, set all bits to 1 (valid).  
If filtering by SPID is performed, trace of the enabled SPID is acquired in all modes regardless of the debug target mode selected by ex\_socunitinfo2 (EX\_UNIT\_HEGESEL).

An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if called during the execution of a user program with the E2 extended function enabled.

An error (EX\_SE\_SFTTRCRUN) is issued if the user program is called during execution (in LPD trace mode) while the external soft trace single-line output function is enabled while the E2 extension function is disabled. return.  
For details on the external soft trace single-wire output function, refer to "Operation Specifications for External Soft Trace Single-Line Output Function".  
  
If called by an MCU that does not support the SPID filtering function, an error (EX\_DE\_SELTRCNOMOUNT) is returned.  
Whether the SPID filtering function is supported can be determined by funcinfo [0]-bit9 of ex\_rh\_getenv.

[Precautions when the initial stop / standby mode debug environment is enabled]

(1) While the user program is running, in the initial stop and standby mode states other than the CyclicRun mode, control of the debug module cannot be performed, so an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed. However, when ICUM is valid, an error (EX\_SE\_STANDBY\_NOSPRT) is returned and processing is not performed even in CyclicRun mode.

[Precautions when supporting multi-core]

(1) The same conditions are set for all cores

(2) ICUM core is not filtered because it does not support this function

#### (2) External soft trace single line output function

1)外部ソフトトレース単線出力機能を有効/無効に設定/参照

Enable / disable external soft trace single-wire output function

ex\_rh\_sfttrcsw

2)外部ソフトトレース単線出力動作状態を設定/参照

2) Set / reference external soft trace single-wire output operating status

ex\_rh\_sfttrcmode

3)外部ソフトトレース単線出力のイベント条件を設定/参照

3) Set / reference event condition of external soft trace single line output

ex\_rh\_sfttrcevent

4)外部ソフトトレース単線出力のフィルタ条件を設定/参照

4) Set / reference filter condition for external soft trace single line output

ex\_rh\_sfttrcfilter

5)ソフトトレースモジュール(TCU\_SFT)の調整機能

5) Adjustment function of soft trace module (TCU\_SFT)

ex\_rh\_sfttrctune

6) 外部ソフトトレース単線出力の記録動作を設定/参照

6) Set / reference the recording operation of external soft trace single line output

ex\_rh\_sfttrcrun

E2拡張機能が使用不可時のみ使用可能な関数

Functions that can be used only when the E2 extension is not available

7)外部ソフトトレース単線出力生データ用メモリの動作状態を設定／参照

7) Set / reference the operating status of the external soft trace single-wire output raw data memory

ex\_rh\_sfttrcrawmemmode

E2拡張機能が使用不可時のみ使用可能な関数

Functions that can be used only when the E2 extension is not available

8)外部ソフトトレース単線出力生データリードに必要な情報を得る

8) Obtain information necessary for external data trace single wire output raw data read

ex\_rh\_sfttrcrawmeminfo

E2拡張機能が使用不可時のみ使用可能な関数

Functions that can be used only when the E2 extension is not available

9)外部ソフトトレース単線出力生データリード

9) External soft trace single wire output raw data read

ex\_rh\_sfttrcrawmemread

E2拡張機能が使用不可時のみ使用可能な関数

Functions that can be used only when the E2 extension is not available

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

外部ソフトトレース単線出力機能は使用できないため、外部ソフトトレース単線出力機能に関連するEXEC I/Fを発行した場合、エラー(EX\_PE\_NOSPRT\_INITSTOPDBG)を返す。

[Notes when the initial stop / standby mode debug environment is enabled]  
Since the external soft trace single-line output function cannot be used, an error (EX\_PE\_NOSPRT\_INITSTOPDBG) is returned when an EXEC I / F related to the external soft trace single-line output function is issued.

[仮想マシン対応時の注意点(G3世代)]

外部ソフトトレース単線出力機能において、仮想マシン対応時に設定が必要な項目はない。仮想マシンでの動作時には、自動的に仮想マシンおよびハードウェアスレッドの外部ソフトトレース単線出力機能に対応する。

[Notes on virtual machine support (G3 generation)]  
In the external soft trace single line output function, there are no items that need to be set when dealing with virtual machines. When operating on a virtual machine, it automatically supports the external soft trace single-line output function of the virtual machine and hardware thread.

[仮想マシン対応時の注意点(G4世代)]

CPU動作モードごとに外部ソフトトレース単線出力有無の指定が可能となる。外部ソフトトレース単線出力有無の指定はデバッグ対象の指定(ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESEL)と連動させる。

[Precautions for virtual machine support (G4 generation)]  
It is possible to specify whether or not to output the external soft trace single line for each CPU operation mode. Specifying whether external soft trace single-line output is used or not is linked with the specification of the debug target (unit\_type = EX\_UNIT\_HEGESEL in ex\_socunitinfo2).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcsw | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力機能のコア毎の有効/無効を設定/参照する。

Sets / references the valid / invalid for each core of the external soft trace single-wire output function.

[Format]

int ex\_rh\_sfttrcsw( USHORT func, ULONG \*enable )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Argument identifier | attribute | Explanation | 対象Target |
| USHORT | func | <I > | Operation mode | ○ |
| ULONG | \*enable | <IO> | 外部ソフトトレース単線出力機能の有効Enable external soft trace single line output function | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定Set |
|  | EX\_REF | 参照Get |
| switch | 外部ソフトトレース単線出力機能の有効をコア毎に指定Enables the external soft trace single line output function for each core | |
|  | bit0 | 0：PE0は無効PE0 is invalid  1：PE0は有効PE0 is valid |
|  | bit1 | 0：PE1は無効  1：PE1は有効 |
|  | bit2 | 0：PE2は無効  1：PE2は有効 |
|  | bit3 | 0：PE3は無効  1：PE3は有効 |
|  | bit4 | 0：PE4は無効  1：PE4は有効 |
|  | bit5 | 0：PE5は無効  1：PE5は有効 |
|  | bit6 | 0：PE6は無効  1：PE6は有効 |
|  | bit7 | 0：PE7は無効  1：PE7は有効 |
|  | bit[31:8] | Unused (0 fixed) |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_DE\_SELSFTTRCNOMOUNT 選択されている外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_EXCLUSION 設定値が排他でない

EX\_PE\_INVALIDCORESELECT 指定したPE番号は現在、無効のため選択できない

EX\_DE\_ALREADYSFTCORESET 外部ソフトトレース単線出力機能の出力対象コア指定が既に設定済みのため

再設定できません。

[Return Values]  
EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_DE\_SELSFTTRCNOMOUNT The selected external soft trace single line output function is not implemented.  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_SE\_CANTCALL\_ASPRUNNING During user program execution using the extended function  
EX\_SE\_SFTTRCRUN Running user program with external soft trace single line output  
EX\_PE\_EXCLUSION Setting value is not exclusive  
EX\_PE\_INVALIDCORESELECT The specified PE number is currently invalid and cannot be selected  
EX\_DE\_ALREADYSFTCORESET Because the output target core specification of the external soft trace single line output function has already been set  
Cannot be reset.

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET：enableで指定した値を設定する。

EX\_REF：現在の設定をenableに返す。

(2)enable

外部ソフトトレース単線出力機能(TCU\_SFT)の有効/無効を各コアのビット単位で設定/参照する。Default値は下記の通り。

・TCU\_SFTがいずれのコアにも実装されていない場合は全て無効(0x00000000)

・TCU\_SFTが実装されている一番若いPE番号のコアが有効(全コアにTCU\_SFTが実装され、PE1が一番若い場合は0x00000002)

[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Set the value specified by enable.  
EX\_REF: Returns the current setting to enable.  
  
(2) enable  
Enables / disables the external soft trace single-wire output function (TCU\_SFT) in bit units of each core. Default values are as follows.  
・ If TCU\_SFT is not implemented in any core, all are disabled (0x00000000)  
・ The core with the smallest PE number in which TCU\_SFT is implemented is valid (0x00000002 if TCU\_SFT is implemented in all cores and PE1 is the youngest)

[Description of operation]

外部ソフトトレース単線出力機能の各コアの有効／無効を設定／参照する。

外部ソフトトレース単線出力機能がサポートされていないときに、いずれかのコアを有効に設定した場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

マルチコアデバイスであっても、複数コアを同時に有効に設定することはできない。もし、複数コアを同時に有効に設定した場合は、エラー(EX\_PE\_EXCLUSION)を返す。

デバイスに実装されていないコアを有効に設定した場合は、エラー(EX\_PE\_INVALIDCORESELECT)を返す。

TCU\_SFTが実装されていないコアを有効に設定した場合は、エラー(EX\_DE\_SELSFTTRCNOMOUNT)を返す。

外部ソフトトレース単線出力機能の有効コア設定は、起動後から以下に示す設定可能期間のみとする。

[設定可能期間]外部ソフトトレース単線出力機能の有効設定し、初めてのプログラム実行開始前までの期間

外部ソフトトレース単線出力機能の有効コア設定を上記の設定可能期間に1度設定した後に異なる設定値を

再設定した場合はエラー(EX\_DE\_ALREADYSFTCORESET)を返す。

[Description of operation]  
Enable / disable each core of external soft trace single line output function.  
  
An error (EX\_DE\_SFTTRCNOMOUNT) is returned if any core is enabled when the external soft trace single line output function is not supported.  
  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
  
Even a multi-core device cannot enable multiple cores simultaneously. If multiple cores are enabled at the same time, an error (EX\_PE\_EXCLUSION) is returned.  
  
An error (EX\_PE\_INVALIDCORESELECT) is returned if a core that is not installed in the device is enabled.  
An error (EX\_DE\_SELSFTTRCNOMOUNT) is returned when a core that does not have TCU\_SFT is enabled.  
  
The effective core setting of the external soft trace single-wire output function is limited to the settable period shown below after startup.  
[Setting period] Period before the first program execution start after valid setting of the external soft trace single-wire output function  
After setting the effective core setting of the external soft trace single line output function once in the above settable period, set different values.  
An error (EX\_DE\_ALREADYSFTCORESET) is returned when resetting.

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能のサポート情報は、ex\_rh\_getenv – funcinfo[6]を参照のこと。また、コア毎のサポート情報は、ex\_rh\_getenv – mcuinfo[6]を参照のこと。

(2)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[Notes for debuggers]  
(1) Refer to ex\_rh\_getenv – funcinfo [6] for the support information of the external soft trace single line output function. Refer to ex\_rh\_getenv – mcuinfo [6] for support information for each core.  
(2) If called when the external soft trace single-wire output function is disabled, no error will occur. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcmode | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力の動作条件を設定/参照する。

Sets / references the operating condition of external soft trace single line output.

[Format]

int ex\_rh\_sfttrcmode( USHORT func, ULONG \*cycle, ULONG \*mode, ULONG \*cond )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性  attribute | 説明Explanation | 対象 |
| USHORT | func | <I > | 操作モードOperation mode | ○ |
| ULONG | \*cycle | <IO> | 外部ソフトトレース単線出力のサイクルExternal soft trace single line output cycle | ○ |
| ULONG | \*mode | <IO> | 外部ソフトトレース単線出力のモードExternal soft trace single line output mode | ○ |
| ULONG | \*cond | <IO> | 外部ソフトトレース単線出力の取得条件External soft trace single line output acquisition conditions | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 set |
|  | EX\_REF | 参照 get |
| \*cycle | | |
|  | EX\_REALTIME\_MODE | リアルタイムトレースモード(Default)  Real-time trace mode (Default) |
|  | EX\_FULL\_MODE | ノンリアルタイムトレース(完全トレース)モード  ※ ノンリアルタイムトレースモードに設定不可の場合がある  EX\_FULL\_MODE Non-real time trace (full trace) mode \* Non-real-time trace mode may not be set. |
| \*mode | | |
|  | EX\_SEC\_TRC | セクション・トレース(イベント検出による自動開始／停止)  ※ 本開発バージョンではサポートしない  Section trace (automatic start / stop by event detection) \* Not supported in this development version |
|  | EX\_ALL\_TRC | 全トレース(Default)  All traces (Default) |
| \*cond | 取得条件をビット単位で指定  Specify acquisition conditions in bit units | |
|  | bit0 | NMで実行されるDBCP命令のトレース出力  Trace output of DBCP instruction executed by NM  0：無効 disabled  1：有効 (Default) Enabled (Default)  ※ 無効に設定不可(有効固定)の場合がある  \* There may be cases where it cannot be set to invalid (fixed to valid). |
|  | bit1 | NMで実行されるDBTAG命令のトレース出力  Trace output of DBTAG instruction executed by NM  0：無効  1：有効 (Default)  ※ 無効に設定不可(有効固定)の場合がある  \* There may be cases where it cannot be set to invalid (fixed to valid). |
|  | bit2 | NMで実行されるDBPUSH命令のトレース出力  0：無効  1：有効 (Default)  Trace output of DBPUSH instruction executed by NM 0: Invalid/Disable 1: Enabled (Default) |
|  | bit3 | VMで実行されるDBCP命令のトレース出力(G3世代のみ。G4世代は0固定)  0：無効  1：有効 (Default)  Trace output of DBCP instruction executed by VM (G3 generation only. G4 generation is fixed to 0) 0: Invalid 1: Enabled (Default) |
|  | bit4 | VMで実行されるDBTAG命令のトレース出力(G3世代のみ。G4世代は0固定)  0：無効  1：有効 (Default)  Trace output of DBTAG instruction executed by VM (G3 generation only. G4 generation is fixed to 0) 0: Invalid 1: Enabled (Default) |
|  | bit5 | VMで実行されるDBPUSH命令のトレース出力(G3世代のみ。G4世代は0固定)  0：無効  1：有効 (Default)  Trace output of DBPUSH instruction executed in VM (G3 generation only. G4 generation is fixed to 0) 0: Invalid 1: Enabled (Default) |
|  | bit6 | DBCP命令の実行時のPC値の出力  0：無効  1：有効 (固定)  ※ 1以外の設定値は無視される。  PC value output during execution of DBCP instruction 0: Invalid 1: Enabled (fixed) \* Settings other than 1 are ignored. |
|  | bit7 | DBTAG命令の実行時のPC値の出力  0：無効  1：有効 (Default)  PC value output when executing DBTAG instruction 0: Invalid 1: Enabled (Default) |
|  | bit8 | DBPUSH命令の実行時のPC値の出力  0：無効  1：有効 (Default)  PC value output during execution of DBPUSH instruction 0: Invalid 1: Enabled (Default) |
|  | bit[31:9] | Unused (0 fixed)  Unused (fixed to 0) |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_SFTTRCNOSPT サポートされていない外部ソフトトレース単線出力の機能なので設定できない

[Return Values]

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_SE\_CANTCALL\_ASPRUNNING During user program execution using the extended function  
EX\_SE\_SFTTRCRUN Running user program with external soft trace single line output  
EX\_PE\_SFTTRCNOSPT Cannot be set because it is an unsupported external soft trace single line output function

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET：各パラメータで指定した値を設定する。

EX\_REF：現在の設定を各パラメータに返す。

(2)cycle

外部ソフトトレース単線出力の動作モードを設定/参照する。

EX\_REALTIME\_MODE ：リアルタイムトレースモード

(ユーザプログラムの実行中にリアルタイムにソフトトレースを観測するモード)

EX\_FULL\_MODE ：ノンリアルタイムトレースモード

(ユーザプログラムが実行するすべてのソフトトレースを観測するモード)

(3)mode

外部ソフトトレース単線出力のモードを設定/参照する。

EX\_SEC\_TRC ：セクション・トレース (本開発バージョンではサポートしない)

(自動開始/停止イベント条件は、ex\_rh\_sfttrceventで設定)

EX\_ALL\_TRC ：全トレース

(4)cond

外部ソフトトレース単線出力の各命令に対する取得条件を設定/参照する。

[Description]  
[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Set the value specified by each parameter.  
EX\_REF: Returns the current setting to each parameter.  
  
(2) cycle  
Sets / references the operation mode of the external soft trace single line output.  
EX\_REALTIME\_MODE: Real-time trace mode  
(Mode for observing soft traces in real time during user program execution)  
EX\_FULL\_MODE: Non-real time trace mode  
(Mode for observing all soft traces executed by the user program)  
  
(3) mode  
Sets / references the mode of external soft trace single line output.  
EX\_SEC\_TRC: Section trace (not supported in this development version)  
(Auto start / stop event conditions are set with ex\_rh\_sfttrcevent)  
EX\_ALL\_TRC: All traces  
  
(4) cond  
Sets / references the acquisition condition for each instruction of external soft trace single line output.

[Description of operation]

外部ソフトトレース単線出力機能の動作条件、取得条件を設定/参照する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

cycleにEX\_REALTIME\_MODEを指定した場合は、オーバーフロー(ロスト)が発生する場合がある。その場合は、オーバーフローが発生したことを示すエラーに関するソフトトレースメッセージが格納される。

cycleにEX\_FULL\_MODEを指定した場合は、トレースメッセージの記録動作をユーザプログラム実行より優先するため、オーバーフローの発生は低減する。

modeにEX\_SEC\_TRCを指定した場合は、ex\_rh\_sfttrceventで指定されたイベント条件を設定する。

modeにEX\_ALL\_TRCを指定した場合は、ex\_rh\_sfttrceventで指定されたイベント条件は無視する。

※本開発バージョンでは、EX\_SEC\_TRCの指定はサポートしない。

[Description of operation]  
Sets / references the operating conditions and acquisition conditions of the external soft trace single-wire output function.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
When EX\_REALTIME\_MODE is specified for cycle, overflow (lost) may occur. In that case, a soft trace message regarding an error indicating that an overflow has occurred is stored.  
When EX\_FULL\_MODE is specified for cycle, the trace message recording operation takes priority over user program execution, so the occurrence of overflow is reduced.  
  
When EX\_SEC\_TRC is specified for mode, the event condition specified by ex\_rh\_sfttrcevent is set.  
When EX\_ALL\_TRC is specified for mode, the event condition specified by ex\_rh\_sfttrcevent is ignored.  
\* EX\_SEC\_TRC specification is not supported in this development version.

[マルチコア対応時の注意点]

(1)現在選択されている物理CPU(PE)に対して設定する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

(2)コアによっては、ノンリアルタイムトレースモードをサポートしない場合がある。また、同様にイベントによる自動開始/停止をサポートしない場合がある。さらに、NMモードで実行されるDBCP命令とDBTAG命令のトレース出力は無効に設定できない場合がある。サポート情報は、ex\_rh\_getenv – funcinfo[8]～[10]を参照のこと。サポートしないコアに対して設定した場合は、エラー(EX\_PE\_SFTTRCNOSPT)を返す。なお、同一デバイス上であっても、サポートコア／未サポートコアが混在する場合がある。

[Notes for debuggers]  
(1) Calling when the external soft trace single-wire output function is disabled will not cause an error. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.  
(2) Some cores may not support the non-real-time trace mode. Similarly, automatic start / stop by event may not be supported. Furthermore, the trace output of DBCP and DBTAG instructions executed in NM mode may not be disabled. Refer to ex\_rh\_getenv – funcinfo [8] to [10] for support information. If it is set for an unsupported core, an error (EX\_PE\_SFTTRCNOSPT) is returned. Note that even on the same device, supported cores / unsupported cores may coexist.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcevent **[This function is not supported]** [This function is not supported] | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力のイベント条件を設定/参照する。

Sets / references the event condition of external soft trace single line output.

[This function is not supported]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcfilter | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力のフィルタ条件を設定/参照する。

Sets / references the filter condition of external soft trace single line output.

[Format]

int ex\_rh\_sfttrcfilter( USHORT func, ULONG filter\_type, ULONG \*num, ULONG \*filter\_info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性 | 説明 | 対象 |
| USHORT | func | <I > | Operation mode | ○ |
| ULONG | filter\_type | <I > | Filter type | ○ |
| ULONG | \*num | <IO> | Number of information by filter | ○ |
| ULONG | \*filter\_info | <IO> | Information by filter | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| func | | | | |
|  | EX\_SET | 設定 | | |
|  | EX\_REF | 参照 | | |
| filter\_type | | | | |
|  | EX\_FILTERTYPE\_PC | PC filtering setting / reference | | |
|  | EX\_FILTERTYPE\_TAG | TAG filtering setting / reference | | |
|  | EX\_FILTERTYPE\_REG | General register filtering setting / reference | | |
|  | EX\_FILTERTYPE\_SPID | SPID filtering setting / reference | | |
| \*num | | | | |
|  | <00000000h-FFFFFFFFh> | | Set the number of filter\_info. \* When referring, the number of returned filter\_info is returned. It is assumed that there are enough filter\_info areas for the specified filter type. Note that \* num must be set and called for all filter types even when referencing. | |
| \*filter\_info | | | | |
|  | <EX\_FILTERTYPE\_PC> (num=2)  \* Not supported in this development version | | | |
|  | filter\_info[0] | PC value to be traced (common to NM / VM) <00000000h-FFFFFFFFh> Default value: 00000000h | | |
|  | filter\_info[1] | PC mask value to be traced (common to NM / VM) <00000000h-FFFFFFFFh> 0: Do not mask (PC value valid) 1: Mask (PC value invalid) Default value: FFFFFFFFh | | |
|  | <EX\_FILTERTYPE\_TAG> (num=3) | | | |
|  | filter\_info[0] | TAG value to trace (common to NM / VM) <00000000h-000003FFh> Default value: 00000000h | | |
|  | filter\_info[1] | TAG mask value to trace (common to NM / VM) <00000000h-000003FFh> 0: Do not mask (TAG value valid) 1: Mask (TAG value invalid) Default value: 000003FFh | | |
|  | filter\_info[2] | TAG filtering mode EX\_RH\_SFTTRC\_FILTERTAG\_SINGLE: Single filtering (fixed) EX\_RH\_SFTTRC\_FILTERTAG\_GROUP: Group filtering \* EX\_RH\_SFTTRC\_FILTERTAG\_GROUP is not supported in this development version. Fixed to EX\_RH\_SFTTRC\_FILTERTAG\_SINGLE. | | |
|  |  |  | | |
|  | <EX\_FILTERTYPE\_REG> (num=1) | | | |
|  | filter\_info[0] | bit0 | | Trace output of general-purpose register r0 0: Invalid 1: Enabled (Default) |
|  |  | bit1 | | Trace output of general-purpose register r1 0: Invalid 1: Enabled (Default) |
|  |  | ・  ・  ・ | |  |
|  |  | bit31 | | Trace output of general-purpose register r31 0: Invalid 1: Enabled (Default) |
|  | <EX\_FILTERTYPE\_SPID> (num=1) | | | |
|  | filter\_info[0] | bit0 | | SPID0 trace output 0: Invalid 1: Enabled (Default) |
|  |  | bit1 | | SPID1 trace output 0: Invalid 1: Enabled (Default) |
|  |  |  | |  |
|  |  | bit31 | | SPID31 trace output 0: Invalid 1: Enabled (Default) |
|  |  |  | | |

[Return Values]  
EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_SE\_CANTCALL\_ASPRUNNING During user program execution using the extended function  
EX\_SE\_SFTTRCRUN Running user program with external soft trace single line output  
EX\_PE\_SFTTRCNOSPT Cannot be set because it is an unsupported external soft trace single line output function

[Description]  
Common to each filter\_type  
[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Set the value specified by each parameter.  
EX\_REF: Returns the current setting to each parameter.  
  
(2) num  
Set the value determined by each filter\_type. Also, secure the filter\_info [] area for the set value.  
  
  
<Filter\_type = EX\_FILTERTYPE\_PC> \* Not supported in this development version

[Argument Description]  
(1) filter\_info [0]  
Set / reference the PC value to be traced.  
(2) filter\_info [1]  
Set / reference the mask value of the PC to be traced.

[Description of operation]  
Set / reference the PC range to be soft-traced by setting the PC value and PC mask value.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
PC filtering settings apply to DBCP, DBTAG, and DBPUSH instructions. However, DBTAG instruction and DBPUSH instruction are not applied when ex\_rh\_sfttrcmode is set to disable PC value output during execution.  
To output trace for all PC values, set filter\_info [1] = FFFFFFFFh.

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[Notes for debuggers]  
(1) Calling when the external soft trace single-wire output function is disabled will not cause an error. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

<Filter\_type = EX\_FILTERTYPE\_TAG>  
  
[Argument Description]  
(1) filter\_info [0]  
Set / reference the TAG value to trace.  
(2) filter\_info [1]  
Sets / references the mask value of the TAG to be traced.  
(3) filter\_info [2]  
Sets / references TAG filtering mode.

[Description of operation]  
Set / reference TAG range to be soft traced by setting TAG value and TAG mask value.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
The TAG filtering setting is applied to the DBTAG instruction. However, this is excluded when group filtering is set.  
To output a soft trace for all TAG values, set filter\_info [1] = 000003FFh.  
  
The operations of single filtering and group filtering are as follows.  
Single filtering: If the filtering results do not match, the soft trace message by executing the DBTAG instruction is not acquired. After that, the filtering result is not affected when executing the soft trace instruction (DBCP instruction, DBPUSH instruction) to be executed.  
Group filtering: If the filtering results do not match, the soft trace message by executing the DBTAG instruction is not acquired. Even when a soft trace instruction is executed thereafter, a uniform soft trace message is not acquired regardless of the filtering result corresponding to each instruction type. This effect continues until the next DBTAG instruction is executed.  
\* The group filtering function is not supported in this development version.

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[Notes for debuggers]  
(1) An error will not occur if this function is called when the external soft trace single-wire output function is disabled. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.  
(2) Some cores may not support group filtering. See ex\_rh\_getenv – funcinfo [11] for support information. If it is set for an unsupported core, an error (EX\_PE\_SFTTRCNOSPT) is returned. Note that even on the same device, supported cores / unsupported cores may coexist.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

＜filter\_type=EX\_FILTERTYPE\_REG＞

[Argument description]

(1)filter\_info[0]

トレースする汎用レジスタをビット単位で設定/参照する。

[Argument Description]  
(1) filter\_info [0]  
Set / reference the general register to be traced in bit units.

[動作説明]

汎用レジスタをビット単位で設定してソフトトレースされる汎用レジスタを設定/参照する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

汎用レジスタフィルタリングの設定は、DBPUSH命令に適用される。

全ての汎用レジスタのソフトトレースを出力する場合は、filter\_info[0]=FFFFFFFFhを設定する。

[Description of operation]  
Set / reference the general register to be soft traced by setting the general register in bit units.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
General register filtering settings apply to the DBPUSH instruction.  
Set filter\_info [0] = FFFFFFFFh to output soft traces for all general-purpose registers.

[マルチコア対応時の注意点]

(1)現在選択されている物理CPU(PE)に対して設定する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[Notes for debuggers]  
(1) An error will not occur if this function is called when the external soft trace single-wire output function is disabled. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

＜filter\_type=EX\_FILTERTYPE\_SPID＞

[Argument description]

(1)filter\_info[0]

フィルタリングするSPIDをビット単位で設定/参照する。

[Argument Description]  
(1) filter\_info [0]  
Set / reference SPID to be filtered in bit units.

[動作説明]

【G3世代】

エラー（EX\_PE\_DATA）を返す。

[Description of operation]  
[G3 generation]  
Return an error (EX\_PE\_DATA).

【G4世代】

ソフトトレースされるSPIDをビット単位で設定/参照する。

特定のSPIDによるフィルタリングを行わない場合は、すべてのビットに1(有効)をセットする。

SPIDによるフィルタリングを行うと、ex\_socunitinfo2(EX\_UNIT\_HEGESEL)で選択したデバッグ対象のモードに関わらず、すべてのモードで、有効にしたSPIDのトレースを取得する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

SPIDフィルタリング機能をサポートしていないMCUで呼び出された場合は、エラー(EX\_PE\_SFTTRCNOSPT)を返す。

SPIDフィルタリング機能をサポートしているかどうかはex\_rh\_getenvのfuncinfo[6] – bit1で判断できる。

[G4 generation]  
Set / reference the SPID to be soft traced in bit units.  
When filtering by a specific SPID is not performed, set all bits to 1 (valid).  
When filtering by SPID is performed, traces of SPIDs enabled in all modes are acquired regardless of the debug target mode selected by ex\_socunitinfo2 (EX\_UNIT\_HEGESEL).  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.  
When called by an MCU that does not support the SPID filtering function, an error (EX\_PE\_SFTTRCNOSPT) is returned.  
Whether or not the SPID filtering function is supported can be determined by funcinfo [6] – bit1 of ex\_rh\_getenv.

[マルチコア対応時の注意点]

(1)全コアに対して同じ条件が設定される

(2)ICUMコアは本機能に対応していないため、フィルタリングされない

[Notes on multi-core support]  
(1) The same conditions are set for all cores  
(2) Since ICUM core does not support this function, it is not filtered

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[Notes for debuggers]  
(1) An error will not occur if this function is called when the external soft trace single-wire output function is disabled. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrctune | **SYNC** | **EMU** | **○** | **BRK** | **○** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **○** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

ソフトトレースの調整機能。

Soft trace adjustment function.

[Format]

int ex\_rh\_sfttrctune( USHORT func, ULONG tune\_type, ULONG \*num, ULONG \*tune\_info )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード  Operation mode | ○ |
| ULONG | tune\_type | <I > | 調整機能種別  Adjustment function type | ○ |
| ULONG | \*num | <IO> | 調整機能別の情報の個数  Number of information by adjustment function | ○ |
| ULONG | \*tune\_info | <IO> | 調整機能別の情報  Information by adjustment function | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |  |
| --- | --- | --- | --- |
| func | | | |
|  | EX\_SET | 設定 | |
|  | EX\_REF | 参照 | |
| tune\_type | | | |
|  | EX\_TUNETYPE\_RATE | 外部I/Fの送信ボーレートを設定/参照  Sets / references the transmission baud rate of the external I / F | |
|  | EX\_TUNETYPE\_OUTPUT | トレース出力の経路と形式を設定/参照  Set / reference trace output path and format | |
|  | EX\_TUNETYPE\_FORMAT | 通信フレームのフォーマットを設定/参照  Set / reference communication frame format | |
| \*num | | | |
|  | <00000000h-FFFFFFFFh> | | tune\_infoの個数を設定する。  ※ 参照時は返したtune\_info数を返す。指定した調整機能種別に足りるtune\_info数の領域があることが前提。なお、参照時でもすべての調整機能種別で、\*numは設定して呼び出すこと。  Set the number of tune\_info. \* Returns the number of tune\_info returned when referring. It is assumed that there are enough tune\_info areas for the specified adjustment function type. Note that \* num must be set and called for all adjustment function types even when referenced. |
| \*tune\_info | | | |
|  | <EX\_TUNETYPE\_RATE> (num=1) | | |
|  | tune\_info[0]  (TRS\_CTRL-EOR) | 外部I/Fの送信ボーレート  00000000h：設定禁止  00000001h：CPUクロック周波数[Hz]x1/2 (Default)  00000002h：CPUクロック周波数[Hz]x1/3  00000003h：CPUクロック周波数[Hz]x1/4  ・  ・  ・  000000FFh：CPUクロック周波数[Hz]x1/256  00000100h～FFFFFFFFh：設定禁止  External I / F transmission baud rate 00000000h: Setting prohibited 00000001h: CPU clock frequency [Hz] x1 / 2 (Default) 00000002h: CPU clock frequency [Hz] x 1/3 00000003h: CPU clock frequency [Hz] x 1/4 ・ ・ ・ 000000FFh: CPU clock frequency [Hz] x 1/256 00000100h to FFFFFFFFh: Setting prohibited | |
|  | <EX\_TUNETYPE\_OUTPUT> (num=2) | | |
|  | tune\_info[0]  (TRS\_CTRL-TOEN) | トレース経路  00000000h：STCLK、STDATA による専用端子出力 (設定禁止)  00000001h：STCLK、STDATA によるLPD-I/F からの直接出力 (設定禁止)  00000002h：LDU経由によるLPD-I/F からの間接出力 (Default)  00000003h～FFFFFFFFh：設定禁止 | |
|  | tune\_info[1]  (TRS\_CTRL-TOSEL) | TCLKとSTDATA によるシリアル出力の転送形式  00000000h：通常UART 出力 (Default)  00000001h：マンチェスターコード出力 (設定禁止)  00000002h～FFFFFFFFh：設定禁止  Trace route 00000000h: Dedicated pin output by STCLK and STDATA (Setting prohibited) 00000001h: Direct output from LPD-I / F by STCLK and STDATA (Setting prohibited) 00000002h: Indirect output from LPD-I / F via LDU (Default) 00000003h to FFFFFFFFh: Setting prohibited Serial output transfer format using TCLK and STDATA 00000000h: Normal UART output (Default) 00000001h: Manchester code output (setting prohibited) 00000002h to FFFFFFFFh: Setting prohibited | |
|  |  |  | |
|  | <EX\_TUNETYPE\_FORMAT> (num=2) | | |
|  | tune\_info[0]  (TRS\_CTRL-FFS) | 通信フレームのストップ・フィールドのフォーマット  00000000h：2ビット (Default)  00000001h：1ビット  00000002h～FFFFFFFFh：設定禁止  Communication frame stop field format 00000000h: 2 bits (Default) 00000001h: 1 bit 00000002h to FFFFFFFFh: Setting prohibited | |
|  | tune\_info[1]  (TRS\_CTRL-FFP) | 通信フレームのパリティ・フィールドのフォーマット  00000000h：パリティなし (Default)  00000001h：奇数パリティ  00000002h：偶数パリティ  00000003h～FFFFFFFFh：設定禁止  Parity field format of communication frame 00000000h: No parity (Default) 00000001h: Odd parity 00000002h: Even parity 00000003h to FFFFFFFFh: Setting prohibited | |
|  |  |  | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_SE\_CANTCALL\_ASPRUNNING During user program execution using the extended function  
EX\_SE\_SFTTRCRUN Running user program with external soft trace single line output

[Description]

各tune\_type共通

Common to each tune\_type

[Argument description]

(1)func

操作モードを指定する。

EX\_SET：各パラメータで指定した値を設定する。

EX\_REF：現在の設定を各パラメータに返す。

(2)num

それぞれのtune\_typeで決められた値を設定する。また、設定した値のtune\_info[]の領域を確保すること。

[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Set the value specified by each parameter.  
EX\_REF: Returns the current setting to each parameter.  
  
(2) num  
Set the value determined by each tune\_type. Also, ensure the tune\_info [] area for the set value.

＜tune\_type=EX\_TUNETYPE\_RATE＞

[Argument description]

(1)tune\_info[0]

外部I/Fの送信ボーレートを設定/参照する。

[Argument Description]  
(1) tune\_info [0]  
Sets / references the transmission baud rate of the external I / F.

[動作説明]

外部I/Fの送信ボーレートを設定/参照する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

[Description of operation]  
Sets / references the transmission baud rate of the external I / F.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.

[マルチコア対応時の注意点]

(1)現在選択されている物理CPU(PE)に対して設定する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).  
  
[Notes for debuggers]  
(1) Calling when the external soft trace single-wire output function is disabled will not cause an error. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.  
  
[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

＜tune\_type= EX\_TUNETYPE\_OUTPUT＞

[Argument description]

(1)tune\_info[0]

トレース出力の経路を設定/参照する。

(2)tune\_info[1]

トレース出力の形式を設定/参照する。

[Argument Description]  
(1) tune\_info [0]  
Set / reference the trace output path.  
(2) tune\_info [1]  
Sets / references the trace output format.

[動作説明]

トレース出力の経路と形式を設定/参照する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

[Description of operation]  
Sets / references the trace output path and format.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.

[マルチコア対応時の注意点]

(1)現在選択されている物理CPU(PE)に対して設定する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).  
  
[Notes for debuggers]  
(1) An error will not occur if this function is called when the external soft trace single-wire output function is disabled. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.  
  
[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

＜tune\_type= EX\_TUNETYPE\_FORMAT＞

[Argument description]

(1)tune\_info[0]

通信フレームのストップ・フィールドのフォーマットを設定/参照する。

(2)tune\_info[1]

通信フレームのパリティ・フィールドのフォーマットを設定/参照する。

[Argument Description]  
(1) tune\_info [0]  
Set / reference the format of the stop field of the communication frame.  
(2) tune\_info [1]  
Sets / references the format of the parity field of the communication frame.

[動作説明]

通信フレームのフォーマットを設定/参照する。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。

[Description of operation]  
Sets / references communication frame format.  
  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
An error (EX\_SE\_CANTCALL\_ASPRUNNING) is returned if this function is called during user program execution with the E2 extended function enabled.  
If the external soft trace single-wire output function is enabled while the E2 extended function is disabled, an error (EX\_SE\_SFTTRCRUN) will be displayed if the user program is called during execution (in LPD trace mode). return.

[マルチコア対応時の注意点]

(1)現在選択されている物理CPU(PE)に対して設定する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能が無効に設定されているときに呼び出した場合にも、エラーとはならない。設定値は、外部ソフトトレース単線出力機能が有効に設定された時に反映される。なお、外部ソフトトレース単線出力の有効／無効の設定は、ex\_socunitinfo2(EX\_UNIT\_TRCSEL)で行う。

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on multi-core support]  
(1) Set for the currently selected physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).  
  
[Notes for debuggers]  
(1) An error will not occur if this function is called when the external soft trace single-wire output function is disabled. The set value is reflected when the external soft trace single line output function is enabled. Note that ex\_socunitinfo2 (EX\_UNIT\_TRCSEL) is used to enable / disable external soft trace single-line output.  
  
[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcrun **[This function is not supported]** [This function is not supported] | **SYNC** | **EMU** | **○** | **BRK** | **×** | **TRC** | **○** | **TIM** | **○** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力の記録動作を設定/参照する。

Sets / references the recording operation of external soft trace single line output.

[Format]

int ex\_rh\_sfttrcrun( USHORT func, ULONG \*run )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モードOperation mode | ○ |
| ULONG | \*run | <IO> | 外部ソフトトレース単線出力の起動/停止External soft trace single line output start / stop | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
| \*run | | |
|  | EX\_SFTTRC\_RUN | 外部ソフトトレース単線出力起動External soft trace single line output start |
|  | EX\_SFTTRC\_STOP | 外部ソフトトレース単線出力停止External soft trace single line output stop |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASPENABLE ASPが有効

EX\_DE\_SFTTRCDISABLE 外部ソフトトレース単線出力が無効

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_USRPGMSTOP ユーザプログラムブレーク中

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_SFTTRACEON 外部ソフトトレース単線出力が起動中なのに起動を指定した

EX\_SE\_SFTTRACEOFF 外部ソフトトレース単線出力が停止中なのに停止を指定した

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASPENABLE ASP is enabled  
EX\_DE\_SFTTRCDISABLE External soft trace single line output is disabled  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_SE\_USRPGMSTOP During user program break  
EX\_SE\_USRPGMRUN User program running  
EX\_SE\_SFTTRACEON Start was specified even though external soft trace single line output is starting  
EX\_SE\_SFTTRACEOFF Stop is specified even though the external soft trace single line output is stopped

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET：runで指定した値を設定する。

EX\_REF：現在の設定をrunに返す。

(2)run

外部ソフトトレース単線出力の起動/停止を設定/参照する。

[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Sets the value specified by run.  
EX\_REF: Returns the current setting to run.  
  
(2) run  
Sets / references start / stop of external soft trace single line output.

[動作説明]

外部ソフトトレース単線出力機能が有効なときの、ユーザプログラム実行中(LPDトレースモード中)の起動/停止を設定／参照する。

E2エミュレータ使用時でかつE2拡張機能が無効の場合に使用できる。E2拡張機能が有効の場合は、エラー(EX\_SE\_ASPENABLE)を返す。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

外部ソフトトレース単線出力機能が無効に設定されている場合は、エラー(EX\_DE\_SFTTRCDISABLE)を返す。

ブレーク中の場合は、エラー(EX\_SE\_USRPGMSTOP)を返す。

外部ソフトトレース単線出力機能が無効に設定されているときに、ユーザプログラムが実行中(通常モード中)で呼び出された場合は、エラー(EX\_SE\_USRPGMRUN)を返す。同様に、ホットプラグインRUN中(接続～強制ブレークするまで)は、エラー(EX\_SE\_USRPGMRUN)を返す。

設定されている状態と同様の設定を行った場合は、エラー(EX\_SE\_SFTTRACEON/EX\_SE\_SFTTRACEOFF)を返す。

[Description of operation]  
Sets / references starting / stopping during user program execution (LPD trace mode) when the external soft trace single-wire output function is enabled.  
  
This can be used when using the E2 emulator and the E2 extension function is disabled. If the E2 extension is enabled, an error (EX\_SE\_ASPENABLE) is returned.  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
If the external soft trace single-wire output function is disabled, an error (EX\_DE\_SFTTRCDISABLE) is returned.  
If a break is in progress, an error (EX\_SE\_USRPGMSTOP) is returned.  
If the external soft trace single-wire output function is disabled, and the user program is called during execution (in normal mode), an error (EX\_SE\_USRPGMRUN) is returned. Similarly, an error (EX\_SE\_USRPGMRUN) is returned during hot plug-in RUN (from connection to forced break).  
An error (EX\_SE\_SFTTRACEON / EX\_SE\_SFTTRACEOFF) is returned when the same setting as the set status is performed.

[マルチコア対応時の注意点]

(1)外部ソフトトレース単線出力用メモリは、共通に使用されるため、物理CPU(PE)の選択に関係ない。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core support]  
(1) Since the external soft trace single-line output memory is used in common, it is not related to the selection of the physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcrawmemmode | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **×** | **TIM** | **×** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力生データ用メモリの動作状態を設定／参照する。

Sets / references the operating status of the external soft trace single-wire output raw data memory.

[Format]

int ex\_rh\_sfttrcrawmemmode( USHORT func, ULONG \*fullstop )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モードOperation mode | ○ |
| ULONG | \*fullstop | <IO> | 外部ソフトトレース単線出力終了モードExternal soft trace single line output end mode | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定 |
|  | EX\_REF | 参照 |
|  |  |  |
| \*fullstop | | |
|  | EX\_NON\_STOP | トレースを終了しない(default)  Do not end tracing (default) |
|  | EX\_FULL\_STOP | 外部ソフトトレース単線出力用メモリがフルで終了  External soft trace single line output memory is full |
|  | EX\_FULL\_BRK | 外部ソフトトレース単線出力用メモリがフルでブレーク  External soft trace single line output memory is full and breaks |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASPENABLE ASPが有効

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_SE\_USRPGMRUN During execution of user program

EX\_FE\_EMUUNKNOWN エミュレータで不明なエラーが発生した

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASPENABLE ASP is enabled  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because it is in asynchronous debug mode  
EX\_SE\_USRPGMRUN User program running  
EX\_FE\_EMUUNKNOWN An unknown error occurred in the emulator

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET：fullstopで指定したトレース動作条件に設定する。

EX\_REF：現在設定されているトレース動作条件をfullstopに返す。

(2)fullstop

トレース・データを記録動作の条件を指定する。

EX\_NON\_STOP ：外部ソフトトレース単線出力用メモリがフル(満杯)になってもトレース・データの記録動作を終了せず、記録動作を継続する。(リングモード/上書きモード)

EX\_FULL\_STOP：外部ソフトトレース単線出力用メモリがフル(満杯)になるとトレース・データの記録動作を終了する。

EX\_FULL\_BRK ：外部ソフトトレース単線出力用メモリがフル(満杯)になるとトレース・データの記録動作を終了して、ユーザプログラムのブレーク要求を発行する。

[Description]  
[Argument Description]  
(1) func  
Specify the operation mode.  
EX\_SET: Sets the trace operating condition specified by fullstop.  
EX\_REF: Returns the currently set trace operation condition to fullstop.  
  
(2) fullstop  
Specify the conditions for recording trace data.  
EX\_NON\_STOP: Even if the external soft trace single-line output memory becomes full (full), the trace data recording operation is not terminated and the recording operation is continued. (Ring mode / Overwrite mode)  
EX\_FULL\_STOP: Stops the trace data recording operation when the external soft trace single-line output memory is full.  
EX\_FULL\_BRK: When the external soft trace single-line output memory is full (full), terminates the trace data recording operation and issues a user program break request.

[動作説明]

外部ソフトトレース単線出力用メモリの記録条件を設定／参照する。

E2エミュレータ使用時でかつE2拡張機能が無効の場合に使用できる。E2拡張機能が有効の場合は、エラー(EX\_SE\_ASPENABLE)を返す。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

ユーザプログラムが実行中に呼び出された場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

不明な原因でエミュレータがエラー応答した場合は、エラー応答(EX\_FE\_EMUUNKNOWN)する。

[Description of operation]  
Sets / references the recording condition of the external soft trace single line output memory.  
  
This can be used when using the E2 emulator and the E2 extension function is disabled. If the E2 extension is enabled, an error (EX\_SE\_ASPENABLE) is returned.  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
If the user program is called during execution, an error (EX\_SE\_USRPGMRUN) is returned.  
If the emulator responds with an error for an unknown reason, it responds with an error (EX\_FE\_EMUUNKNOWN).

[マルチコア対応時の注意点]

(1)外部ソフトトレース単線出力用メモリは、共通に使用されるため、物理CPU(PE)の選択に関係ない。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core support]  
(1) Since the external soft trace single-line output memory is used in common, it is not related to the selection of the physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[デバッガ関係者向け注意事項]

(1)外部ソフトトレース単線出力機能の設定変更を行うEXEC I/Fを発行した場合であっても、外部ソフトトレース単線出力用メモリはクリアされない。

[Notes for debuggers]  
(1) The memory for external soft trace single line output is not cleared even when an EXEC I / F that changes the setting of the external soft trace single line output function is issued.

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcrawmeminfo | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **×** | **TIM** | **×** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力生データリードに必要な情報を取得。

Acquires information necessary for reading raw data of external soft trace single wire.

[Format]

int ex\_rh\_sfttrcrawmeminfo( ULONG \* maxsize )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性 | 説明 | 対象 |
| ULONG \* | maxsize | < O> | 外部ソフトトレース単線出力用メモリに格納されている生データフレーム数  Number of raw data frames stored in external soft trace single line output memory | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| maxsize | | |
|  |  | トレースされている生データフレーム数  Number of raw data frames being traced |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASPENABLE ASPが有効

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_DE\_M\_TRCMEMOPENFAILED トレースメモリのオープンに失敗した

EX\_DE\_M\_TRCMEMACCFAILED トレースメモリアクセスに失敗した

EX\_SE\_USRPGMRUN During execution of user program

EX\_FE\_EMUUNKNOWN エミュレータで不明なエラーが発生した

EX\_DE\_SFTCANTREC　　　　 外部ソフトトレース単線出力機能は記録できない状態を継続中

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASPENABLE ASP is enabled  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because it is in asynchronous debug mode  
EX\_DE\_M\_TRCMEMOPENFAILED Trace memory open failed  
EX\_DE\_M\_TRCMEMACCFAILED Trace memory access failed  
EX\_SE\_USRPGMRUN User program running  
EX\_FE\_EMUUNKNOWN An unknown error occurred in the emulator  
EX\_DE\_SFTCANTREC The external soft trace single line output function is in a state where it cannot be recorded.

[Description]

[Argument description]

(1)maxsize

外部ソフトトレース単線出力用メモリに格納されている有効フレーム数を返す。

[Description]  
[Argument Description]  
(1) maxsize  
Returns the number of valid frames stored in the external soft trace single line output memory.

[動作説明]

外部ソフトトレース単線出力用メモリに格納されている有効フレーム数を返す。

E2エミュレータ使用時でかつE2拡張機能が無効の場合に使用できる。E2拡張機能が有効の場合は、エラー(EX\_SE\_ASPENABLE)を返す。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

ユーザプログラムが実行中に呼び出された場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

トレース・データが1フレームもない場合、maxsize=0としてエラー応答はしない。

[Description of operation]  
Returns the number of valid frames stored in the external soft trace single line output memory.  
This can be used when using the E2 emulator and the E2 extension function is disabled. If the E2 extension is enabled, an error (EX\_SE\_ASPENABLE) is returned.  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
If the user program is called during execution, an error (EX\_SE\_USRPGMRUN) is returned.  
If there is no trace data, maxsize = 0 and no error response is made.

何らかの原因で外部ソフトトレース単線出力生データ用メモリが開くことができなかった場合は、エラー応答(EX\_DE\_M\_TRCMEMOPENFAILED)する。また、何らかの原因で外部ソフトトレース単線出力生データの情報が取得できなかった場合は、エラー応答(EX\_DE\_M\_TRCMEMACCFAILED)する。更に、不明な原因でエミュレータがエラー応答した場合は、エラー応答(EX\_FE\_EMUUNKNOWN)する。

If the external soft trace single line output raw data memory cannot be opened for some reason, an error response (EX\_DE\_M\_TRCMEMOPENFAILED) is returned. If the external soft trace single line output raw data information cannot be acquired for some reason, an error response (EX\_DE\_M\_TRCMEMACCFAILED) is returned. Furthermore, if the emulator responds with an error for an unknown reason, an error response (EX\_FE\_EMUUNKNOWN) is made.

外部ソフトトレース単線出力機能の記録動作が正常に行えないことを通知するため、

「外部ソフトトレース単線出力機能の記録が正常に行えない条件」が成立した場合は、

エラー(EX\_DE\_SFTCANTREC)を返す。このエラーは、ex\_exitまでエラーを継続する。

In order to notify that the recording operation of the external soft trace single line output function cannot be performed normally,  
If the “Condition that recording of the external soft trace single line output function cannot be performed normally” is satisfied,  
An error (EX\_DE\_SFTCANTREC) is returned. This error continues until ex\_exit.

[外部ソフトトレース単線出力機能の記録が正常に行えない条件]

　　以下①～③すべての条件が成立した場合とする。

* 1. LPD 1pin
  2. 外部ソフトトレース単線出力機能を使用
  3. 以下のいずれかのブレーク機能が発生

es\_expfuncact\_brk 拡張機能のアクションブレーク

　　　 es\_expfuncfull\_bk 拡張機能のフルブレーク

　　　es\_sfttrcfull\_bk 外部ソフトトレース単線出力フルブレーク

　　　es\_enf\_bk 強制ブレーク

[Conditions when the external soft trace single-wire output function cannot be recorded normally]  
 Suppose the following conditions are met.  
① LPD 1pin  
② Use external soft trace single line output function  
③ One of the following break functions occurs  
es\_expfuncact\_brk Extension action break  
     Es\_expfuncfull\_bk Extended function full break  
      Es\_sfttrcfull\_bk External soft trace single line output full break  
      Es\_enf\_bk forced break

[マルチコア対応時の注意点]

(1)外部ソフトトレース単線出力用メモリは、共通に使用されるため、物理CPU(PE)の選択に関係ない。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core support]  
(1) Since the external soft trace single-line output memory is used in common, it is not related to the selection of the physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[デバッガ関係者向け注意事項]

(1)ex\_rh\_sfttrcrawmemreadで指定する値は、本関数のmaxsizeの値を超えないようにする必要がある。

(2)外部ソフトトレース単線出力生データリード時のフレームとは、トレース・データの解析が行える単位(トレースメッセージ)ではなく、あくまで外部ソフトトレース単線出力用メモリへの格納単位であり注意が必要である。

1トレース・データの解析に必要なフレーム数は、トレースメッセージによって可変であり、1フレーム～3フレームが必要となる。(トレースメッセージ先頭フラグおよびトレースパケット内のトレースタイプを確認することで必要フレーム数を判断する。)

(3)外部ソフトトレース単線出力機能の設定変更を行うEXEC I/Fを発行した場合であっても、外部ソフトトレース単線出力用メモリはクリアされない。

(4) 外部ソフトトレース単線出力機能は記録できない状態になった場合は、EXECは、エラー(EX\_DE\_SFTCANTREC)を返す。

デバッガは、このエラーを受け取った後、ユーザに以下のガイドを行う。

「外部ソフトトレース単線出力機能を使用する場合は、デバッガを再起動してください」

[Notes for debuggers]  
(1) The value specified by ex\_rh\_sfttrcrawmemread must not exceed the maxsize value of this function.  
(2) External soft trace single-line output Raw data read frame is not a unit that can be used to analyze trace data (trace message), but is a unit that is stored in the external soft trace single-line output memory. is there.  
The number of frames required for analyzing 1 trace data is variable depending on the trace message, and 1 to 3 frames are required. (The required number of frames is determined by checking the trace message head flag and the trace type in the trace packet.)  
(3) The memory for external soft trace single line output is not cleared even when an EXEC I / F that changes the setting of the external soft trace single line output function is issued.  
(4) The EXEC returns an error (EX\_DE\_SFTCANTREC) when the external soft trace single-line output function becomes unable to record.  
After receiving this error, the debugger guides the user as follows.  
"Restart the debugger when using the external soft trace single line output function"

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_sfttrcrawmemread | **SYNC** | **EMU** | **×** | **BRK** | **○** | **TRC** | **×** | **TIM** | **×** | **COV** | **／** | **HOT** | **×** |
| **ASYNC** | **EMU1** | **×** | **EMU2** | **×** | **BRK1** | **×** | **BRK2** | **×** | **TRC1** | **×** | **TRC2** | **×** |

[Function]

外部ソフトトレース単線出力生データリード

External soft trace single wire output raw data read

[Format]

int ex\_rh\_sfttrcrawmemread( ULONG sfttrcrawmemaddr, ULONG sfttrcrawmemunm, UCHAR \* sfttrcrawmembuf )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | Argument identifier | 属性 | 説明 | 対象 |
| ULONG | sfttrcrawmemaddr | <I > | 読み出す外部ソフトトレース単線出力生データフレームの先頭アドレス  Start address of external soft trace single line output raw data frame to be read | ○ |
| ULONG | sfttrcrawmemunm | <I > | 読み出す外部ソフトトレース単線出力生データフレーム数  Number of external soft trace single-wire output raw data frames to be read | ○ |
| UCHAR \* | sfttrcrawmembuf | < O> | 外部ソフトトレース単線出力生データを返すバッファのポインタ  Buffer pointer to return external soft trace single wire output raw data | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| sfttrcrawmemaddr | | |
|  |  | 読み出す外部ソフトトレース単線出力生データフレームの先頭アドレス  Start address of external soft trace single line output raw data frame to be read |
|  |  |  |
| sfttrcrawmemunm | | |
|  |  | 読み出す外部ソフトトレース単線出力生データフレーム数  Number of external soft trace single-wire output raw data frames to be read |
|  |  |  |
| \*sfttrcrawmembuf | | |
|  |  | 外部ソフトトレース単線出力生データを返すバッファのポインタ |
|  |  | Buffer pointer to return external soft trace single wire output raw data |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_DE\_SFTTRCNOMOUNT 外部ソフトトレース単線出力機能は実装されていない

EX\_SE\_ASPENABLE ASPが有効

EX\_SE\_ASYNCMODE\_ENB 非同期デバッグモード中のため設定/参照/削除できない

EX\_DE\_M\_TRCMEMOPENFAILED トレースメモリのオープンに失敗した

EX\_DE\_M\_TRCMEMACCFAILED トレースメモリアクセスに失敗した

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NOTRACEBLOCK トレースが見つからない

EX\_FE\_EMUUNKNOWN エミュレータで不明なエラーが発生した

EX\_DE\_SFTCANTREC　　　　 外部ソフトトレース単線出力機能は記録できない状態を継続中

EX\_NOERROR Successful completion  
EX\_FE\_COMM EXEC is not running  
EX\_PE\_DATA setting value is abnormal  
EX\_DE\_SFTTRCNOMOUNT External soft trace single line output function is not implemented  
EX\_SE\_ASPENABLE ASP is enabled  
EX\_SE\_ASYNCMODE\_ENB Cannot set / reference / delete because of asynchronous debug mode  
EX\_DE\_M\_TRCMEMOPENFAILED Trace memory open failed  
EX\_DE\_M\_TRCMEMACCFAILED Trace memory access failed  
EX\_SE\_USRPGMRUN User program running  
EX\_SE\_NOTRACEBLOCK Trace not found  
EX\_FE\_EMUUNKNOWN An unknown error occurred in the emulator  
EX\_DE\_SFTCANTREC The external soft trace single line output function is in a state where it cannot be recorded.

[Description]

[Argument description]

(1)sfttrcrawmemaddr

読み出す外部ソフトトレース単線出力生データフレームの先頭アドレスを指定する。

外部ソフトトレース単線出力生データフレームのアドレスは、外部ソフトトレース単線出力用メモリに格納された最古のデータを0としたアドレスになる。

(2)sfttrcrawmemunm

読み出す外部ソフトトレース単線出力生データフレーム数を指定する。

(3)sfttrcrawmembuf

外部ソフトトレース単線出力生データを返すバッファのポインタを指定する。

sfttrcrawmembufには、sfttrcrawmemnum×16byteのデータが格納される。

[Argument Description]  
(1) sfttrcrawmemaddr  
Specifies the start address of the external soft trace single-wire output raw data frame to be read.  
The address of the external soft trace single line output raw data frame is an address in which the oldest data stored in the external soft trace single line output memory is 0.  
(2) sfttrcrawmemunm  
Specifies the number of external soft trace single-wire output raw data frames to be read.  
(3) sfttrcrawmembuf  
Specifies the pointer of the buffer that returns the external soft trace single-wire output raw data.  
In sfttrcrawmembuf, data of sfttrcrawmemnum × 16 bytes is stored.

[動作説明]

sfttrcrawmemaddrで指定されたフレームからsfttrcrawmemnum個分の外部ソフトトレース単線出力生データをsfttrcrawmembufに格納する。

E2エミュレータ使用時でかつE2拡張機能が無効の場合に使用できる。E2拡張機能が有効の場合は、エラー(EX\_SE\_ASPENABLE)を返す。

外部ソフトトレース単線出力機能がサポートされていない場合は、エラー(EX\_DE\_SFTTRCNOMOUNT)を返す。

ユーザプログラムが実行中に呼び出された場合は、エラー(EX\_SE\_USRPGMRUN)を返す。

トレース・データが一つもない場合は、エラー応答(EX\_SE\_NOTRACEBLOCK)する。

何らかの原因で外部ソフトトレース単線出力生データ用メモリが開くことができなかった場合は、エラー応答(EX\_DE\_M\_TRCMEMOPENFAILED)する。また、何らかの原因で外部ソフトトレース単線出力生データをリードできなかった場合は、エラー応答(EX\_DE\_M\_TRCMEMACCFAILED)する。更に、不明な原因でエミュレータがエラー応答した場合は、エラー応答(EX\_FE\_EMUUNKNOWN)する。

外部ソフトトレース単線出力機能の記録動作が正常に行えないことを通知するため、

「外部ソフトトレース単線出力機能の記録が正常に行えない条件」が成立した場合は、

エラー(EX\_DE\_SFTCANTREC)を返す。このエラーは、ex\_exitまでエラーを継続する。

[Description of operation]  
Stores sfttrcrawmemnum external soft trace single-line output raw data from the frame specified by sfttrcrawmemaddr in sfttrcrawmembuf.  
  
This can be used when using the E2 emulator and the E2 extension function is disabled. If the E2 extension is enabled, an error (EX\_SE\_ASPENABLE) is returned.  
If the external soft trace single line output function is not supported, an error (EX\_DE\_SFTTRCNOMOUNT) is returned.  
If the user program is called during execution, an error (EX\_SE\_USRPGMRUN) is returned.  
If there is no trace data, an error response (EX\_SE\_NOTRACEBLOCK) is returned.  
  
If the external soft trace single line output raw data memory cannot be opened for some reason, an error response (EX\_DE\_M\_TRCMEMOPENFAILED) is returned. If the external soft trace single-wire output raw data cannot be read for some reason, an error response (EX\_DE\_M\_TRCMEMACCFAILED) is returned. Furthermore, if the emulator responds with an error for an unknown reason, an error response (EX\_FE\_EMUUNKNOWN) is made.  
  
In order to notify that the recording operation of the external soft trace single line output function cannot be performed normally,  
If the “Condition that recording of the external soft trace single line output function cannot be performed normally” is satisfied,  
An error (EX\_DE\_SFTCANTREC) is returned. This error continues until ex\_exit.

[外部ソフトトレース単線出力機能の記録が正常に行えない条件]

　　以下①～③すべての条件が成立した場合とする。

1. LPD 1pin
2. 外部ソフトトレース単線出力機能を使用
3. 以下のいずれかのブレーク機能が発生

es\_expfuncact\_brk 拡張機能のアクションブレーク

　　　 es\_expfuncfull\_bk 拡張機能のフルブレーク

　　　es\_sfttrcfull\_bk 外部ソフトトレース単線出力フルブレーク

　　　es\_enf\_bk 強制ブレーク

[Conditions when the external soft trace single-wire output function cannot be recorded normally]  
  Suppose the following conditions are met.  
① LPD 1pin  
② Use external soft trace single line output function  
③ One of the following break functions occurs  
es\_expfuncact\_brk Extension action break  
      Es\_expfuncfull\_bk Extended function full break  
       Es\_sfttrcfull\_bk External soft trace single line output full break  
       Es\_enf\_bk forced break

[マルチコア時の注意点]

(1)外部ソフトトレース単線出力用メモリは、共通に使用されるため、物理CPU(PE)の選択に関係ない。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2(EX\_UNIT\_COREPESEL)で行う。

[Notes on multi-core]  
(1) Since the external soft trace single-line output memory is used in common, it is not related to the selection of the physical CPU (PE).  
(2) The physical CPU (PE) is switched by ex\_socunitinfo2 (EX\_UNIT\_COREPESEL).

[デバッガ開発者向け注意事項]

(1)外部ソフトトレース単線出力生データリード時のフレームとは、トレース・データの解析が行える単位(トレースメッセージ)ではなく、あくまで外部ソフトトレース単線出力用メモリへの格納単位であるため、注意が必要である。

1トレース・データの解析に必要なフレーム数は、トレースメッセージによって可変であり、1フレーム～3フレームが必要となる。(トレースメッセージ先頭フラグおよびトレースパケット内のトレースタイプを確認することで必要フレーム数を判断する。)

外部ソフトトレース単線出力用メモリ使用時 ：16byte/1フレーム

[Notes for debugger developers]  
(1) External soft trace single line output Raw data reading frame is not a unit that can analyze trace data (trace message) but a unit that is stored in the external soft trace single line output memory. is necessary.  
The number of frames required for analyzing 1 trace data is variable depending on the trace message, and 1 to 3 frames are required. (The necessary number of frames is determined by checking the trace message head flag and the trace type in the trace packet.)  
When using external soft trace single line output memory: 16 bytes / frame

表 3‑21 外部ソフトトレース単線出力用メモリ使用時の1フレームの構成

Table 3-21 Configuration of one frame when using external soft trace single-line output memory

|  |  |  |  |
| --- | --- | --- | --- |
| No. | 1byte目～8byte目のbit[7:1]  bit [7: 1] of byte 1 to byte 8 | 8byte目のbit0  Bit0 of byte 8 | 9byte目～16byte目  Byte 9 to byte 16 |
| 1 | リザーブデータ  Reserve data | トレースメッセージ先頭フラグ  (1：トレースメッセージ先頭  0：トレースメッセージ先頭でない)  Trace message head flag (1: Trace message head 0: Not the beginning of the trace message) | トレースパケット  Trace packet |

(2)外部ソフトトレース単線出力機能の設定変更を行うEXEC I/Fを発行した場合でも、外部ソフトトレース単線出力用メモリのトレース・データをクリアすることはない。

(2) Even when an EXEC I / F that changes the setting of the external soft trace single line output function is issued, the trace data in the external soft trace single line output memory is not cleared.

(3)ex\_rh\_sfttrcrawmemreadでリードするトレースメッセージのフォーマットは「G3世代 トレース・コントロール・ユニット(TCU)機能仕様書」「G4世代 トレース・コントロール・ユニット(TCU)機能仕様書」を参照のこと。なお、外部ソフトトレース単線出力機能の場合でも、トレースメッセージのフォーマットは内蔵トレース機能のソフトトレース命令実行時に出力されるトレースメッセージと同等である。

(3) Refer to “G3 Generation Trace Control Unit (TCU) Functional Specification” and “G4 Generation Trace Control Unit (TCU) Functional Specification” for the format of the trace message read by ex\_rh\_sfttrcrawmemread. Even in the case of the external soft trace single line output function, the format of the trace message is the same as the trace message output when executing the soft trace instruction of the internal trace function.

(4)内蔵トレース機能のトレースメッセージと異なり、外部ソフトトレース単線出力では出力されるPC値はすべてフルアドレスで出力される。(すべてのメッセージが同期化メッセージとして出力される。)

(4) Unlike the trace message of the built-in trace function, all the PC values ​​that are output with the external soft trace single line output are output with the full address. (All messages are output as synchronization messages.)

(5)ソフトトレースメッセージの前には必ずタイムスタンプ同期化メッセージが出力される。

(5) A time stamp synchronization message is always output before the soft trace message.

(6)外部ソフトトレース単線出力機能は記録できない状態になった場合は、EXECは、エラー(EX\_DE\_SFTCANTREC)を返す。

デバッガは、このエラーを受け取った後、ユーザに以下のガイドを行う。

「外部ソフトトレース単線出力機能を使用する場合は、デバッガを再起動してください」

(6) The EXEC returns an error (EX\_DE\_SFTCANTREC) when the external soft trace single-line output function becomes unable to record.  
After receiving this error, the debugger guides the user as follows.  
"Restart the debugger when using the external soft trace single line output function"

[非同期デバッグモード時の注意点]

(1)非同期デバッグモードには対応しない。非同期デバッグモード時に呼び出した場合は、エラー(EX\_SE\_ASYNCMODE\_ENB)を返す。

[Notes on asynchronous debug mode]  
(1) Does not support asynchronous debug mode. If called in asynchronous debug mode, an error (EX\_SE\_ASYNCMODE\_ENB) is returned.

#### カバレッジ関数

本開発バージョンではカバレッジ機能をサポートしない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covmapping **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジの領域設定/参照を行う

[Format]

int ex\_covmapping( USHORT func, ULONG \*saddr, ULONG \*eaddr )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| ULONG | \*saddr | <IO> | 開始アドレス | -- |
| ULONG | \*eaddr | <IO> | 終了アドレス | -- |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| eaddr | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covmapping2 **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジの領域設定/参照を行う

[Format]

int ex\_covmapping2( USHORT func, USHORT \*cov\_num, EXCOVMAP \*cov )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | -- |
| USHORT | \*cov\_num | <IO> | カバレッジ領域情報の個数 | -- |
| EXCOVMAP | \*cov | <IO> | カバレッジ領域情報(複数可能) | -- |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <IO> | 開始アドレス | -- |
|  | ULONG | eaddr | <IO> | 終了アドレス | -- |
|  | ULONG | dummy | < > | 予備 | -- |
| }EXCOVMAP | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| cov\_num | | |
|  | ----- | ----- |
|  |  |  |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| eaddr | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covsw **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジの有効/無効を切り替える。

[Format]

int ex\_covsw( USHORT cov\_mem\_sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | cov\_mem\_sw | <I > | 有効無効情報 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| cov\_mem\_sw | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covrun **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジ測定の開始/終了の切り替えを行う

[Format]

int ex\_covrun( USHORT cov\_sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | cov\_sw | <I > | 開始終了情報 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| cov\_sw | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covclear **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジ測定結果の初期化を行う。

[Format]

int ex\_covclear(USHORT mode, EXCOVINIT \*clear )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 初期化領域選択 | ○ |
| EXCOVINIT | \*clear | <I > | 初期化領域 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <I > | 開始アドレス | ○ |
|  | ULONG | eaddr | <I > | 終了アドレス | ○ |
| }EXCOVINIT | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_no | | |
|  | ----- | ----- |
|  |  |  |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| eaddr | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covsearch **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

指定したアドレス範囲からカバレッジデータを検索する

[Format]

int ex\_covsearch( USHORT mode, EXCOVSCH \*covaddr )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | mode | <I > | 検索方法 | ○ |
| EXCOVSCH | covaddr | <IO> | 検索条件 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <I > | 検索範囲の下限アドレス | ○ |
|  | ULONG | number | <I > | 検索する範囲のアドレス | ○ |
|  | ULONG | schaddr | < O> | 検出アドレス | ○ |
|  | USHORT | count | <I > | 検索データ個数 | ○ |
|  | USHORT | direct | <I > | 検出方向 | ○ |
|  | USHORT | \*data | <I > | 検索データ(複数個可) | ○ |
|  | USHORT | schrslt | < O> | 検索結果(発見または未発見) | ○ |
| }EXCOVSCH | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| mode | | |
|  | ----- | ----- |
|  |  |  |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| number | | |
|  | ----- | ----- |
|  |  |  |
| schaddr | | |
|  | ----- | ----- |
|  |  |  |
| count | | |
|  | ----- | ----- |
|  |  |  |
| direct | | |
|  | ----- | ----- |
|  |  |  |
| data | | |
|  | ----- | ----- |
|  |  |  |
| schrslt | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covread **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジ測定結果を読み出す

[Format]

int ex\_covread( USHORT cov\_flag, USHORT mode, EXCOVRD \*covrd )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | cov\_flag | <I > | 読み出すデータを指定する | ○ |
| USHORT | mode | <I > | 読み出しモードを指定する | ○ |
| EXCOVRD | covrd | <IO> | 読み出し範囲および結果 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <I > | 開始アドレス | ○ |
|  | ULONG | eaddr | <I > | 終了アドレス | ○ |
|  | ULONG | number | < O> | 読み出したバイト数 | ○ |
|  | ULONG | f\_count | < O> | 実行されたフェッチ数 | ○ |
|  | ULONG | r\_count | < O> | 実行されたリード数 | -- |
|  | ULONG | w\_count | < O> | 実行されたライト数 | -- |
|  | ULONG | count | < O> | 実行されたフェッチ数 | ○ |
|  | ULONG | \*data | < O> | カバレッジ測定結果(複数個可) | ○ |
| }EXCOVRD | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| cov\_flag | | |
|  | ----- | ----- |
|  |  |  |
| mode | | |
|  | ----- | ----- |
|  |  |  |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| eaddr | | |
|  | ----- | ----- |
|  |  |  |
| number | | |
|  | ----- | ----- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| f\_count | | |
|  | ----- | ----- |
|  |  |  |
| r\_count | | |
|  | ----- | ----- |
|  |  |  |
| w\_count | | |
|  | ----- | ----- |
|  |  |  |
| count | | |
|  | ----- | ----- |
|  |  |  |
| \*data | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_covwrite **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

カバレッジ測定結果をカバレッジメモリに書き込む

[Format]

int ex\_covwrite( EXCOVWR \*covwr )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXCOVWR | covwr | <I > | 書き込み情報 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | saddr | <I > | 開始アドレス | ○ |
|  | ULONG | eaddr | <I > | 終了アドレス | ○ |
|  | UCHAR | \*data | <I > | 書き込むデータ | ○ |
| }EXCOVWR | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| saddr | | |
|  | ----- | ----- |
|  |  |  |
| eaddr | | |
|  | ----- | ----- |
|  |  |  |
| \*data | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

#### タイマ関数 / パフォーマンス計測関数

1)タイマの有効/無効を設定

ex\_timesw

2)タイマの停止/再開を設定

ex\_timerun

3)タイマの計測結果のクリア

ex\_timeclear

4)タイマの計測結果の参照

ex\_timeread, ex\_timeread2, ex\_rh\_timeread

5)タイマの計測条件の設定/参照/削除

ex\_timeevent2, ex\_timeevent3, ex\_rh\_timeevet

6)タイマの動作モードの設定/参照

ex\_timemode

7)パフォーマンス計測動作の有効/無効を設定

ex\_rh\_performancesw

8)パフォーマンス計測結果のクリア

ex\_rh\_performanceclear

9)パフォーマンス計測結果の参照

ex\_rh\_performanceread

10)パフォーマンス計測条件の設定/参照/削除

ex\_rh\_performanceset

※取り消し線が書かれた関数はRH850ではサポートしない。

＜時間計測/パフォーマンス計測機能概要＞

RH850ではTMUを利用して時間計測、パフォーマンス計測を行う。

サポートする計測機能は大きく分けて以下の4種類ある。

・RUN-BREAKタイマ(V850E2互換)

・TMU\_TCKタイマ

・TMU\_CLKタイマ(V850E2互換)

・パフォーマンス計測機能

各計測機能比較を表 3‑22に示す。

表 3‑22 時間計測/パフォーマンスの計測機能比較

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | RUN-BREAKタイマ | TMU\_TCKタイマ | TMU\_CLKタイマ | パフォーマンス  計測機能 |
| カウンタ本数 | | 1本/コア | 3本/コア | 4本1セット/コア | 4本/コア |
| カウント単位 | | デバッグクロック | デバッグクロック | CPUクロック | ----  (CPUクロック※1) |
| 排他使用 | | なし | なし | あり  TMU\_CLKタイマを使用する場合にパフォーマンス計測は使用不可。  パフォーマンス計測を1chでも使用する場合にTMU\_CLKタイマは使用不可。  互いに計測機能を切り替える場合は計測条件設定関数で削除してから行うこと。 | |
| 計測区間 | | 実行～ブレーク | イベント間  実行～イベント  イベント～ブレーク  実行～ブレーク  から1区間選択 | イベント間 | イベント間  実行～イベント  イベント～ブレーク  実行～ブレーク  から1区間選択 |
| 計測内容 | | 実行時間 | 蓄積時間  パスカウント  最大時間  最小時間  最新時間  から1項目選択 | 蓄積時間  パスカウント  最大時間  最小時間 | ※2 |
| タイマ番号/パフォーマンス番号 | | 0固定  全コア共通 | 0以外  全コア、全計測機能で共通  (各コア/各計測機能で異なる番号を付与) | | |
| 関数 | | | | | |
|  | 計測条件設定 | なし | ex\_rh\_timeevent | ex\_timeevent2 | ex\_rh\_performanceset |
| 計測結果参照 | ex\_timeread2 | ex\_rh\_timeread | ex\_timeread2 | ex\_rh\_performanceread |
| 計測結果クリア | なし | ex\_timeclear | ex\_timeclear | ex\_rh\_performanceclear |
| 計測スイッチ | なし | ex\_timesw | ex\_timesw | ex\_rh\_performancesw |
| 備考 | | V850E2と互換。  デフォルトで設定済み。  計測の停止や計測結果のクリアはできない。  実行開始直前に内部的に計測結果がクリアされる。 | 3本個別に区間と計測項目を設定可能。  デバッグクロックで計測されるため時間の算出にCPUクロックは不要。 | V850E2と互換。  時間の算出にはCPUクロックが必要。  蓄積時間とパスカウントから平均時間を算出可能。 | 4本個別に区間と計測項目を設定可能。  時間計測以外の項目も計測可能。 |

※1 時間計測した場合

※2 パフォーマンス計測の詳細は、表 3‑23を参照

[デバッガ関係者向け注意事項]

PFC1Aはデバイスに不具合があるため、タイマおよびパフォーマンス計測機能はサポートしない。従って、タイマおよびパフォーマンス計測に関連するI/Fを呼び出したときの動作は保証しない。

[仮想マシン対応時の注意点(G3世代)]

1. RUN-BREAKタイマおよびTMU\_TCKタイマは特に設定を変更することなくNMモードとVMモードを区別なく計測可能。
2. TMU\_CLKタイマはVMモード中を除いた実行時間の計測が可能。
3. パフォーマンス計測機能では、NMモードとVMモードを区別なく計測することができないため、NMモードとVMモードの計測結果を足し合わせる必要がある。NMモードとVMモードの設定の詳細は、ex\_rh\_performancesetを参照。

[仮想マシン対応時の注意点(G4世代)]

1. RUN-BREAKタイマ、TMU\_TCKタイマは仮想化支援機能の有効/無効、および、CPU動作モードに関わらず、常時計測される。
2. パフォーマンス計測機能、および、TMU\_CLKタイマでは、仮想化支援機能有効時は、ex\_socunitinfo2のunit\_type=EX\_UNIT\_HEGESELでデバッグ対象に選択したCPU動作モードで実行されている区間の現象のみが計測される。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timesw | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

タイマ動作の有効/無効の切り換えを行う。

[Format]

int ex\_timesw( EXTIMERUN \*sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXTIMERUN | \*sw | <I > | 有効無効情報 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | timer\_sw | <I > | 動作の切り換え | ○ |
|  | USHORT | num | <I > | 切り換えるタイマ数 | ○ |
|  | USHORT | \*timer\_no | <I > | タイマ番号(複数個可) | ○ |
| }EXTIMERUN | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_sw | | |
|  | EX\_TIME\_SW\_OFF | 無効 |
|  | EX\_TIME\_SW\_ON | 有効 |
|  |  |  |
| num | | |
|  | <1 - FFFFh> | 指定するタイマ番号の数 |
|  |  |  |
| \*timer\_no | | |
|  | <0, 1 - FFFFh> | タイマ番号 |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_ALREADY 既に同じ状態になっている

EX\_SE\_NONTIMNO タイマ番号が設定されていない

EX\_SE\_USRPGMRUN During execution of user program

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_NOSUPRT This function is not supported

[Description]

[Argument description]

(1)timer\_sw

動作の切り替えを指定する。

EX\_TIME\_SW\_OFF ：無効(計測開始イベントが発生しても、計測されない)

EX\_TIME\_SW\_ON ：有効

(2)num

有効/無効を切り替えるタイマ数を指定する。

(3)timer\_no

有効/無効を切り替えるタイマ番号を指定する。

numで指定したタイマ数だけ、タイマ番号を指定可能。

[動作説明]

タイマ動作の有効/無効を切り替える。

無効状態のタイマは計測開始イベントが発生しても計測されない。計測を行うには対象のタイマ番号の動作を有効にしておく必要がある。

ブレーク中のみ実行可能で、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)が返る。

存在しないタイマ番号を指定した場合はエラー(EX\_SE\_NONTIMNO)を返す。

既に有効[無効]状態のタイマに対して、有効[無効]に設定しようとした場合はエラー(EX\_SE\_ALREADY)を返す。(デバッガ側ではEX\_SE\_ALREADYが返ることにより、タイマの状態を確認できる)

num = 1, timer\_no = 0ですべてのタイマを指定可能。同期デバッグモードの場合、全PEのタイマが操作対象となるが、非同期デバッグモードの場合は選択PEのタイマのみが操作対象となる。すべてのタイマを指定した場合、EX\_SE\_ALREADYのエラーは返らない。このエラーは無視してすべてのタイマを操作する。num = 1以外でtimer\_no = 0を指定した場合はエラー(EX\_PE\_DATA)が返る。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のタイマ番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。ただし、num = 1, timer\_no = 0で全タイマを指定した場合は、現在選択されている物理CPU(PE)以外の物理CPU(PE)も含むすべてのタイマを切り替えることが可能。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合、タイマの計測結果クリアを行う。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timerun **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

タイマ動作(停止/再開)の切り換えを行う。

[Format]

int ex\_timerun( EXTIMERUN \*sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXTIMERUN | \*sw | <--> |  | -- |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | timer\_sw | <--> |  | -- |
|  | USHORT | num | <--> |  | -- |
|  | USHORT | \*timer\_no | <--> |  | -- |
| }EXTIMERUN | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_sw | | |
|  | ----- | ----- |
|  |  |  |
| num | | |
|  | ----- | ----- |
|  |  |  |
| \*timer\_no | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timeclear | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

タイマの計測結果の初期化を行う。

[Format]

int ex\_timeclear( USHORT timer\_no )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | timer\_no | <I > | タイマ番号 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_no | | |
|  | <0, 1 - FFFFh> | タイマ番号  0は全番号指定 |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXECが起動していない

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONTIMNO タイマ番号が設定されていない

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_NOSUPRT This function is not supported

[Description]

[Argument description]

(1)timer\_no

0 ：すべての設定済みタイマ番号を指定

1-FFFFh ：個別のタイマ番号を指定

[動作説明]

指定したタイマ番号の計測結果を初期化する。

ブレーク中のみ実行可能で、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)が返る。

存在しないタイマ番号を指定した場合はエラー(EX\_SE\_NONTIMNO)が返る。

timer\_no = 0ですべてのタイマ機能を指定可能。同期デバッグモードの場合、全PEのタイマが操作対象となるが、非同期デバッグモードの場合は選択PEのタイマのみが操作対象となる。

RUN-BREAKタイマの計測結果は本関数によりクリアされない。ユーザプログラム実行開始時に内部的にクリアされる。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のタイマ番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。ただし、timer\_no = 0で全タイマを指定した場合は、現在選択されている物理CPU(PE)以外の物理CPU(PE)を含むすべてのタイマの計測結果を初期化することが可能。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timeread **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

タイマの計測結果の参照を行う。

[Format]

int ex\_timeread( USHORT timer\_no, EXTIMERD \*timer )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | timer\_no | <--> |  | -- |
| EXTIMERD | \*timer | <--> |  | -- |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | ULONG | count | <--> |  | -- |
|  | ULONG | timer\_total | <--> |  | -- |
|  | USHORT | clock\_unit | <--> |  | -- |
|  | USHORT | mode | <--> |  | -- |
|  | ULONG | timer\_max | <--> |  | -- |
|  | ULONG | timer\_min | <--> |  | -- |
|  | ULONG | timer\_avg | <--> |  | -- |
| }EXTIMERD | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_no | | |
|  | ----- | ----- |
|  |  |  |
| timer\_total | | |
|  | ----- | ----- |
|  |  |  |
| clock\_unit | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timeread2 | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

タイマの計測結果の参照を行う。

RUN-BREAKタイマとTMU\_CLKタイマの計測結果読み出し関数

[Format]

int ex\_timeread2( USHORT timer\_no, EXTIMERD2 \*timer )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | timer\_no | <I > | タイマ番号 | ○ |
| EXTIMERD2 | \*timer | < O> | 計測結果 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | info | < O> | カウンタの状態 | ○ |
|  | USHORT | dummy | <--> |  | -- |
|  | ULONG | clock\_unit | < O> | 1カウントの時間単位 | ○ |
|  | ULONG | count\_l | < O> | パスカウント(区間通過回数) | ○ |
|  | ULONG | count\_h | <--> |  | -- |
|  | ULONG | timer\_total\_l | < O> | 積算カウント | ○ |
|  | ULONG | timer\_total\_h | <--> |  | -- |
|  | ULONG | timer\_max\_l | < O> | 最大カウント | ○ |
|  | ULONG | timer\_max\_h | <--> |  | -- |
|  | ULONG | timer\_min\_l | < O> | 最小カウント | ○ |
|  | ULONG | timer\_min\_h | <--> |  | -- |
| }EXTIMERD2 | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_no | | |
|  | <0, 1 - FFFFh> | タイマ番号  0はRUN-BREAKタイマ結果の参照 |
|  |  |  |
| info | | |
|  | ビット単位でオーバーフロー/閾値違反発生/計測中を示す。 | |
|  | EX\_TIM\_COUNTOVFL | パスカウント(区間通過回数)オーバーフロー |
|  | EX\_TIM\_MAXOVFL | 最大カウントオーバーフロー |
|  | EX\_TIM\_TOTALOVFL | 積算カウントオーバーフロー |
|  | EX\_TIM\_COUNTTVFL | パスカウント(区間通過回数)閾値違反発生 |
|  | EX\_TIM\_MAXTVFL | 最大カウントの閾値違反発生 |
|  | EX\_TIM\_TOTALTVFL | 積算カウントの閾値違反発生 |
|  | EX\_TIM\_MINTVFL | 最小カウントの閾値違反発生 |
|  | EX\_TIM\_COUNTINGNOW | 区間内で計測中 |
|  |  |  |
| clock\_unit | | |
|  | <0, 1 - FFFFFFFFh> | 単位はナノ秒  (0の場合、CPUクロックであることを示す) |
|  |  |  |
| count\_l | | |
|  | <0 - FFFFFFFFh> | パスカウント(区間通過回数) |
|  | | |

|  |  |  |
| --- | --- | --- |
| timer\_total\_l | | |
|  | <0 - FFFFFFFFh> | 積算カウント |
|  | | |
| timer\_max\_l | | |
|  | <0 - FFFFFFFFh> | 最大カウント |
|  | | |
| timer\_min\_l | | |
|  | <0 - FFFFFFFFh> | 最小カウント |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONTIMNO タイマ番号が設定されていない

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_RUNBRKNOSPT 時間計測(RUN-BRK)機能未サポート

EX\_PE\_CPUTIMENOSPT　　　　区間計測(CPUクロック利用)機能未サポート

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

[Argument description]

(1)timer\_no

タイマの計測結果の参照を行うタイマ番号を指定する。

RUN-BREAKタイマは0を指定すること。

ex\_timeevent2()で設定したTMU\_CLKタイマの番号を指定するとTMU\_CLKタイマの計測結果が読み出し可能。存在しないタイマ番号を指定した場合は、エラー(EX\_SE\_NONTIMNO)を返す。

(2)info

タイマの計測結果に対するオーバーフロー/閾値違反発生/計測中の情報をビット単位で返す。

EX\_TIM\_COUNTOVFL ：パスカウントがオーバーフロー

EX\_TIM\_MAXOVFL ：最大カウントがオーバーフロー

EX\_TIM\_TOTALOVFL ：積算カウントがオーバーフロー

EX\_TIM\_COUNTTVFL ：パスカウントが閾値違反発生

EX\_TIM\_MAXTVFL ：最大カウントが閾値違反発生

EX\_TIM\_TOTALTVFL ：積算カウントが閾値違反発生

EX\_TIM\_MINTVFL ：最小カウントが閾値違反発生

EX\_TIM\_COUNTINGNOW ：区間内で計測中

RUN-BREAKタイマでオーバーフローが発生した場合、EX\_TIM\_MAXOVFLとEX\_TIM\_TOTALOVFLを同時に返す。

TMU\_CLKタイマでは、パスカウント、最大カウント、積算カウントがオーバーフローした場合に各bitが1になる。最小カウントはFFFFFFFFh以下のためオーバーフローしない。閾値違反モードで計測した場合に閾値違反が発生すると各bitが1になる。

(3)clock\_unit

1カウントの時間単位を返す。単位はナノ秒。

RUN-BREAKタイマはデバッグクロックで計測するため、その値が返る。TMU\_CLKタイマは0が返る。

0が返る場合は、CPUクロックであることを示す。カウント値から実時間を表示する場合は、CPUクロックをユーザに入力してもらい、算出する必要がある。

(4)count\_l

TMU\_CLKタイマではパスカウントを返す。イベント間の通過回数を示す。そのため、TMU\_CLKタイマに指定した区間を一度も実行しなかった場合は0が返る。

開始イベントが成立後、終了イベントが成立して1カウントとなるため、区間内を実行中にブレークした場合はまだカウントされていない状態となる。

(5)timer\_total\_l

timer\_noに0を指定した場合、RUN-BREAKタイマの結果を返す。

timer\_noでTMU\_CLKタイマを指定した場合、積算カウントが返る。

(6)timer\_max\_l

timer\_noに0を指定した場合、RUN-BREAKタイマの結果を返す。

timer\_noでTMU\_CLKタイマを指定した場合、最大カウントが返る。

(7)timer\_min\_l

timer\_noに0を指定した場合、RUN-BREAKタイマの結果を返す。

timer\_noにTMU\_CLKタイマを指定した場合、最小カウントが返る。

(8)count\_h, timer\_total\_h, timer\_max\_h, timer\_min\_l

カウンタは32bitであるため使用しない。0固定。

[動作説明]

timer\_noで指定したタイマ番号の計測結果を返す。

0指定でRUN-BREAKタイマ、0以外でTMU\_CLKタイマの計測結果を指定する。

RUN-BREAKタイマはデバッグ用クロックでカウント(カウンタ：32bit)する。

デバッグ用クロックが16.5MHzの場合、カウンタ値が最大値(FFFFFFFFh)になりオーバーフローフラグが成立する時間は約4.5分となる。

ユーザプログラムの実行状態がこの時間以上経過すると、0からカウントを再開し、再度カウンタ値が最値大値(FFFFFFFFh)になると、その時点でユーザプログラムの実行状態に関わらず時間計測は停止する。

そのため、本関数を発行してオーバーフローフラグが成立状態でかつカウンタ値が最大値(FFFFFFFFh)の場合は、正常な計測結果ではなくなるので注意すること。

TMU\_CLKタイマはCPUクロックでカウント(カウンタ：32bit)する。

デバッガ等で実行時間を表示する場合は、カウンタ値から実行時間を計算するためにCPUクロックの値が必要になる。ユーザにCPUクロックを入力してもらうなどの対応が必要。

RUN-BREAKタイマの計測結果の読み出しは、ユーザプログラム実行中でもエラー応答しないで、infoに「EX\_TIM\_COUNTINGNOW」を付加、count\_lに「0」、timer\_total\_l、timer\_max\_l、timer\_min\_lに、現在の計測時間を返す。

ただし、E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

また、E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

TMU\_CLKタイマの計測結果の読み出しは、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)を返す。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によって時間計測(RUN-BRK)機能や区間計測(CPUクロック利用)機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すと以下に示すエラーを返す。

時間計測(RUN-BRK)機能が未実装時：EX\_PE\_RUNBRKNOSPTを返す。

区間計測(CPUクロック利用)機能が未実装時：EX\_PE\_CPUTIMENOSPTを返す。

なお、各機能の実装有無は、ex\_rh\_getenvのfuncinfo[12]、funcinfo[14]を参照すること。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のタイマ番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_timeread | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

タイマの計測結果の参照を行う。

TMU\_TCKタイマ専用の計測結果読み出し関数。

[Format]

int ex\_rh\_timeread( USHORT timer\_no, EXRHTIMERD \*timer )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | timer\_no | <I > | タイマ番号 | ○ |
| EXRHTIMERD | \*timer | < O> | 計測結果 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | info | < O> | カウンタの状態 | ○ |
|  | USHORT | type | < O> | 計測項目 | ○ |
|  | ULONG | clock\_unit | < O> | 1カウントの時間単位 | ○ |
|  | ULONG | count | < O> | カウント | ○ |
| }EXRHTIMERD | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| timer\_no | | |
|  | <1 - FFFFh> | タイマ番号 |
|  | | |
| type | | |
|  | EX\_TIM\_TYPEADD | 積算カウント |
|  | EX\_TIM\_TYPEMAX | 最大カウント |
|  | EX\_TIM\_TYPEMIN | 最小カウント |
|  | EX\_TIM\_TYPENEW | 最新カウント |
|  | EX\_TIM\_TYPEPAS | パスカウント(区間通過回数) |
|  | | |
| info | | |
|  | ビット単位でオーバーフロー/閾値違反/計測中を示す | |
|  | EX\_TIM\_OVFL | オーバーフロー |
|  | EX\_TIM\_TVFL | 閾値違反発生 |
|  | EX\_TIM\_COUNTINGNOW | 区間内で計測中 |
|  |  |  |
| clock\_unit | | |
|  | <0, 1 - FFFFFFFFh> | 単位はナノ秒  (0の場合、CPUクロックであることを示す) |
|  |  |  |
| count | | |
|  | <0 - FFFFFFFFh> | カウント値 |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONTIMNO タイマ番号が設定されていない

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_PE\_DCKTIMENOSPT　　　 区間計測(デバッグ用クロック利用)機能未サポート

[Description]

[Argument description]

(1)timer\_no

TMU\_TCKタイマ結果の参照を行うタイマ番号を指定する。

ex\_rh\_timeeventで設定したTMU\_TCKタイマの番号を指定するとTMU\_TCKタイマの計測結果が読み出し可能。存在しないタイマ番号を指定した場合は、エラー(EX\_SE\_NONTIMNO)を返す。

ex\_timeevent2で設定したTMU\_CLKタイマ番号は指定できない。また、0も設定できない。RUN-BREAKタイマの計測結果を参照するときはex\_timeread2を使用すること。

(2)type

計測項目情報を返す。

EX\_TIM\_TYPEADD ：積算カウント

EX\_TIM\_TYPEMAX ：最大カウント

EX\_TIM\_TYPEMIN ：最小カウント

EX\_TIM\_TYPENEW ：最新カウント

EX\_TIM\_TYPEPAS ：パスカウント(区間通過回数)

(3)info

時間計測結果に対するオーバーフロー/閾値違反発生/計測中の情報をビット単位で返す。

EX\_TIM\_OVFL ：オーバーフロー

EX\_TIM\_TVFL ：閾値違反発生

EX\_TIM\_COUNTINGNOW ：区間内で計測中

(4)clock\_unit

1カウントの時間単位を返す。単位はナノ秒。

本関数ではTMU\_TCKタイマ機能のみをサポートするため、0が返ることはない。将来、機能拡張した場合は0が返る可能性もある。0が返る場合は、CPUクロックであることを示す。

(5)count

計測結果が返る。clock\_unitの値を用いて実時間を算出することが可能。

[動作説明]

timer\_noで指定したTMU\_TCKタイマ番号の計測結果を返す。

RUN-BREAKタイマの計測結果は、ex\_timeread2で読み出すこと。

ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)を返す。

設定されていないタイマ番号を設定した場合はエラー(EX\_SE\_NONTIMNO)を返す。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によって区間計測(DCUクロック利用)機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_DCKTIMENOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[13]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のタイマ番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timeevent2 | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

タイマの計測条件の設定/参照/削除を行う。

TMU\_CLKタイマ専用。

[Format]

int ex\_timeevent2( USHORT func, USHORT timer\_no, EXTIMEINF2 \*time )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | timer\_no | <I > | タイマ番号 | ○ |
| EXTIMEINF2 | \*time | <IO> | 計測条件 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <IO> | 動作モード | ○ |
|  | USHORT | statno | <IO> | ----- | -- |
|  | USHORT | \*s\_event | <IO> | 開始条件イベント番号(0終端) | ○ |
|  | USHORT | \*e\_event | <IO> | 終了条件イベント番号(0終端) | ○ |
|  | ULONG | count\_limit\_l | <IO> | パスカウント(閾値) | ○ |
|  | ULONG | count\_limit\_h | <IO> | 積算カウント(閾値) | ○ |
|  | ULONG | timer\_limit\_l | <IO> | 最大カウント(閾値) | ○ |
|  | ULONG | timer\_limit\_h | <IO> | 最小カウント(閾値) | ○ |
| }EXTIMEINF2 | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定(新規作成/変更) |
|  | EX\_REF | 参照 |
|  | EX\_DEL | 削除 |
|  |  |  |
| timer\_no | | |
|  | <1 - FFFFh> | タイマ番号。0は指定不可。 |
|  |  |  |
| mode | | |
|  | EX\_CORE\_CNT\_NORMAL | 通常モード |
|  | EX\_CORE\_CNT\_BREAK | 閾値違反ブレークモード |
|  |  |  |
| statno | | |
|  | ----- | ----- |
|  |  | |

|  |  |  |
| --- | --- | --- |
| \*s\_event | | |
|  | <0, 1 - FFFFh> | 計測開始イベント番号  単数のみ(2個目に0を設定すること) |
|  |  |  |
| \* e\_event | | |
|  | <0, 1 - FFFFh> | 計測終了イベント番号  単数のみ(2個目に0を設定すること) |
|  |  |  |
| count\_limit\_l | | |
|  | <0 - FFFFFFFFh> | mode = EX\_CORE\_CNT\_BREAK時のみ有効 |
|  |  |  |
| count\_limit\_h | | |
|  | <0 - FFFFFFFFh> | mode = EX\_CORE\_CNT\_BREAK時のみ有効 |
|  |  |  |
| timer\_limit\_l | | |
|  | <0 - FFFFFFFFh> | mode = EX\_CORE\_CNT\_BREAK時のみ有効 |
|  |  |  |
| timer\_limit\_h | | |
|  | <0 - FFFFFFFFh> | mode = EX\_CORE\_CNT\_BREAK時のみ有効 |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONTIMNO タイマ番号が存在しない(func = EX\_DEL、EX\_REF時)

EX\_SE\_FULLTIMNO 全タイマが設定済み

EX\_SE\_NONEVNNO イベント番号が存在しない(func = EX\_SET時)

EX\_PE\_INVEVNCOND 設定不可のイベントを指定した

EX\_PE\_ANOTHERPEEVN 別のPEのイベント番号は条件に設定できない

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_NOSUPRT This function is not supported

EX\_SE\_CLKUSEDFORPFM CLKはパフォーマンス機能に使用されているためタイマに使用できない

EX\_PE\_TIMNOUSED タイマ番号はTMU\_CLKタイマ以外で使用済みのため使用できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_CPUTIMENOSPT　　　　区間計測(CPUクロック利用)機能未サポート

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET ：timer\_noで指定したタイマ番号を設定(新規作成/変更)

EX\_REF ：timer\_noで指定したタイマ番号の設定状態を参照

EX\_DEL ：timer\_noで指定したタイマ番号を削除

(2)timer\_no

タイマ番号を指定する。

0は指定不可。

(3)mode

動作モードを設定する。

EX\_OORE\_CNT\_NORMAL ：通常モード

EX\_CORE\_CNT\_BREAK ：閾値違反ブレークモード

(4)statno

未使用。設定値は無視する。参照すると0が返る。

(5)\*s\_event

タイマの計測開始イベントに設定するイベント番号を設定する。

(6)\*e\_event

タイマの計測終了イベントに設定するイベント番号を設定する。

(7)count\_limit\_l

パスカウントの閾値を設定する。

mode = EX\_CORE\_CNT\_BREAKの時のみ有効。他のmodeでは設定を無視する。

(8)count\_limit\_h

積算カウントの閾値を設定する。

mode = EX\_CORE\_CNT\_BREAKの時のみ有効。他のmodeでは設定を無視する。

(9)timer\_limit\_l

最大カウントの閾値を設定する。

mode = EX\_CORE\_CNT\_BREAKの時のみ有効。他のmodeでは設定を無視する。

(10)timer\_limit\_h

最小カウントの閾値を設定する。

mode = EX\_CORE\_CNT\_BREAKの時のみ有効。他のmodeでは設定を無視する。

[動作説明]

イベント2点間の実行時間を計測する条件を設定する。

計測結果の読み出しはex\_timeread2で行う。

TMU\_CLKタイマの計測条件は選択PEに対して設定される。対象のPEに切り替えてから計測条件を設定すること。

TMU\_CLKタイマ番号は0以外を指定すること。また、タイマ番号は全PE間で共通に管理するため、別のPEで使用していないタイマ番号を指定すること。別のPEで使用中のタイマ番号を指定した場合、エラー(EX\_PE\_ANOTHERPETIM)を返す。TMU\_TCKタイマおよびパフォーマンス計測機能とも番号を共通で管理するため、これらの番号と同じタイマ番号も使用できない。使用した場合はエラー(EX\_PE\_TIMNOUSED)を返す。

TMU\_CLKはタイマ以外にパフォーマンス計測を行う際にも同じリソースが使用される。このためパフォーマンス計測機能と排他使用となる。そのため、本関数を呼び出した時点でパフォーマンス計測条件が設定済みの場合、エラー(EX\_SE\_CLKUSEDFORPFM)を返す。

イベント2点間を実行した積算カウント、1回の実行に要した最大カウントと最小カウント、イベント2点間を通過したパスカウントを計測することが可能。イベント2点間の平均実行時間は「積算カウント÷パスカウント」で算出可能。

カウント単位がCPUクロックとなるため、計測時間として表示する場合はユーザにによるCPUクロック設定値の入力が必要。

mode = EX\_CORE\_CNT\_BREAK(閾値違反モード)に設定すると、各計測項目の閾値に設定された値をカウント値がオーバーフローすると、ブレークが発生する。最小カウントはアンダーフローでブレークが発生。閾値違反モードで実行すると、ex\_timeread2で参照しても、計測結果は読み出せない。設定した閾値が読み出される。なお、ブレークはe\_eventで設定したイベントでタイマ計測を終了したときに閾値違反となっていた場合に発生する。(カウント中にカウンタが閾値を超えた場合でも、すぐにブレークは発生しない。)

開始イベント、終了イベントに設定できるイベントは、それぞれex\_rh\_evncondで設定したTEU\_CPU条件のイベント1個、または、ex\_rh\_evnintgcondで設定したTEU\_CPU条件のシーケンシャルイベント1個のみ。それ以外のイベントを設定した場合はエラー(EX\_PE\_INVEVNCOND)を返す。また、開始/終了イベントに複数個のイベントを設定した場合はエラー(EX\_PE\_DATA)を返す。

ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。

(次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に参照可能とする)

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によって区間計測(CPUクロック利用)機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_CPUTIMENOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[14]を参照すること。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合、時間計測結果のクリアを行う。

(2)新規作成したTMU\_CLKタイマはデフォルトで「有効」状態になる。

(3)本関数は、V850E2のタイマ機能と互換性を持たせたI/F仕様である。パフォーマンス計測機能と計測リソースが同じであるため排他で使用すること。パフォーマンス計測機能で1chでも設定済みの場合はエラー(EX\_SE\_CLKUSEDFORPFM)を返す。TMU\_CLKタイマもex\_rh\_performancesetを利用して設定することで、さらに自由度のある(区間や計測項目を個別に設定した)時間計測が可能となる。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)に対して処理される。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定したタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。

(4)開始/終了イベントに設定可能なイベントは現在選択されている物理CPU(PE)で設定されたイベント番号のみ。現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定されたイベント番号を指定した場合はエラー(EX\_PE\_ANOTHERPEEVN)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timeevent3 **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

タイマの計測条件の設定/参照/削除を行う。

[Format]

int ex\_timeevent3( USHORT func, USHORT timer\_no, EXTIMEINF3 \*time )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <--> |  | -- |
| USHORT | timer\_no | <--> |  | -- |
| EXTIMEINF3 | \*time | <--> |  | -- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <--> |  | -- |
|  | USHORT | statno | <--> |  | -- |
|  | USHORT | \*s\_event | <--> |  | -- |
|  | USHORT | \*e\_event | <--> |  | -- |
|  | ULONG | count\_limit\_l | <--> |  | -- |
|  | ULONG | count\_limit\_h | <--> |  | -- |
|  | ULONG | timer\_total\_limit\_l | <--> |  | -- |
|  | ULONG | timer\_total\_limit\_h | <--> |  | -- |
|  | ULONG | timer\_max\_limit\_l | <--> |  | -- |
|  | ULONG | timer\_max\_limit\_h | <--> |  | -- |
|  | ULONG | timer\_min\_limit\_l | <--> |  | -- |
|  | ULONG | timer\_min\_limit\_h | <--> |  | -- |
| }EXTIMEINF3 | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| timer\_no | | |
|  | ----- | ----- |
|  |  |  |
| mode | | |
|  | ----- | ----- |
|  |  |  |
| statno | | |
|  | ----- | ----- |
|  |  |  |
| \*s\_event | | |
|  | ----- | ----- |
|  |  |  |
| \*e\_event | | |
|  | ----- | ----- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| count\_limit\_l | | |
|  | ----- | ----- |
|  |  |  |
| count\_limit\_h | | |
|  | ----- | ----- |
|  |  |  |
| timer\_total\_limit\_l | | |
|  | ----- | ----- |
|  |  |  |
| timer\_totallimit\_h | | |
|  | ----- | ----- |
|  |  |  |
| timer\_max\_limit\_l | | |
|  | ----- | ----- |
|  |  |  |
| timer\_max\_limit\_h | | |
|  | ----- | ----- |
|  |  |  |
| timer\_min\_limit\_l | | |
|  | ----- | ----- |
|  |  |  |
| timer\_min\_limit\_h | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_timeevent | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

タイマの計測条件の設定/参照/削除を行う。

TMU\_TCKタイマ専用。

[Format]

int ex\_rh\_timeevent( USHORT func, USHORT timer\_no, EXRHTIMEINF \*time )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | timer\_no | <I > | タイマ番号 | ○ |
| EXRHTIMEINF | \*time | <IO> | 計測条件 | ○ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <IO> | 動作モード | ○ |
|  | USHORT | type | <IO> | 計測項目 | ○ |
|  | USHORT | s\_event | <IO> | 開始条件イベント番号 | ○ |
|  | USHORT | e\_event | <IO> | 終了条件イベント番号 | ○ |
|  | ULONG | count\_limit | <IO> | カウント(閾値) | ○ |
| }EXRHTIMEINF | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定(新規作成/変更) |
|  | EX\_REF | 参照 |
|  | EX\_DEL | 削除 |
|  |  |  |
| timer\_no | | |
|  | <1 - FFFFh> | タイマ番号。0は指定不可。 |
|  |  |  |
| mode | | |
|  | EX\_CORE\_CNT\_NORMAL | 通常モード |
|  | EX\_CORE\_CNT\_BREAK | 閾値違反ブレークモード |
|  |  |  |
| type | | |
|  | EX\_TIM\_TYPEADD | 積算カウント |
|  | EX\_TIM\_TYPEMAX | 最大カウント |
|  | EX\_TIM\_TYPEMIN | 最小カウント |
|  | EX\_TIM\_TYPENEW | 最新カウント |
|  | EX\_TIM\_TYPEPAS | パスカウント(区間通過回数) |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| s\_event | | |
|  | <0, 1 - FFFFh> | 計測開始イベント番号  0は実行開始で計測も開始 |
|  |  |  |
| e\_event | | |
|  | <0, 1 - FFFFh> | 計測終了イベント番号  0はブレークで計測も停止 |
|  |  |  |
| count\_limit | | |
|  | <0 - FFFFFFFFh> | mode = EX\_CORE\_CNT\_BREAK時のみ有効 |
|  |  |  |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONTIMNO タイマ番号が存在しない(func = EX\_DEL、EX\_REF時)

EX\_SE\_FULLTIMNO 全タイマが設定済み

EX\_SE\_NONEVNNO イベント番号が存在しない(func = EX\_SET時)

EX\_PE\_INVEVNCOND 設定不可のイベントを指定した

EX\_PE\_ANOTHERPEEVN 別のPEのイベント番号は条件に設定できない

EX\_PE\_ANOTHERPETIM 別のPEのタイマ番号は処理できない

EX\_NOSUPRT This function is not supported

EX\_PE\_TIMNOUSED タイマ番号はTMU\_TCKタイマ以外で使用済みのため使用できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_DCKTIMENOSPT　　　 区間計測(デバッグ用クロック利用)機能未サポート

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET ：timer\_noで指定したタイマ番号を設定(新規作成/変更)

EX\_REF ：timer\_noで指定したタイマ番号の設定状態を参照

EX\_DEL ：timer\_noで指定したタイマ番号を削除

(2)timer\_no

タイマ番号を指定する。

0は指定不可。

(3)mode

動作モードを設定する。

EX\_OORE\_CNT\_NORMAL ：通常モード

EX\_CORE\_CNT\_BREAK ：閾値違反ブレークモード

(4)type

計測項目を指定する。

EX\_TIM\_TYPEADD ：積算カウント

EX\_TIM\_TYPEMAX ：最大カウント

EX\_TIM\_TYPEMIN ：最小カウント

EX\_TIM\_TYPENEW ：最新カウント

EX\_TIM\_TYPEPAS ：パスカウント(区間通過回数)

(5)s\_event

タイマの計測開始イベントに設定するイベント番号を設定する。

0を指定した場合は、実行と同時に計測も開始する。

(6)e\_event

タイマの計測終了イベントに設定するイベント番号を設定する。

0を指定した場合は、ブレークと同時に計測も停止する。

(7)count\_limit

カウントの閾値を設定する。

mode = EX\_CORE\_CNT\_BREAKの時のみ有効。他のmodeでは設定を無視する。

[動作説明]

イベント2点間の実行時間を計測する。開始イベント番号に0を指定した場合は、実行と同時に時間計測を開始し、終了イベント番号に0を指定した場合は、ブレークと同時に時間計測を停止する。組み合わせによって、開始イベント～終了、実行～終了イベントまでの計測も可能。ただし、開始条件や終了条件に0を指定した場合、本来RUN-BREAKタイマの計測に用いる指定のため、最大値、最小値、最新値は計測できなく、常に蓄積値が計測されるので注意。また、パスカウントは計測中に終了条件に一致した時点でカウントされるため、終了条件がない場合はカウントされないため注意。開始条件や終了条件に0を指定する使い方は推奨しない。

計測結果の読み出しはex\_rh\_timereadで行う。

TMU\_TCKタイマの計測条件は選択PEに対して設定される。対象のPEに切り替えてから計測条件を設定すること。

TMU\_TCKタイマ番号は0以外を指定すること。また、タイマ番号は全PE間で共通に管理するため、別のPEで使用していないタイマ番号を指定すること。別のPEで使用中のタイマ番号を指定した場合、エラー(EX\_PE\_ANOTHERPETIM)を返す。TMU\_CLKタイマおよびパフォーマンス計測機能とも番号を共通で管理するため、これらの番号と同じタイマ番号も使用できない。使用した場合はエラー(EX\_PE\_TIMNOUSED)を返す。

イベント2点間を実行した積算カウント、1回の実行に要した最大または最小のカウント、直前に実行した最新のカウント、イベント2点間を通過したパスカウントのいずれか1項目を計測することが可能。イベント2点間の平均実行時間は「積算カウント÷パスカウント」で算出可能。

カウント単位はデバッグ用クロックに依存する。

mode = EX\_CORE\_CNT\_BREAK(閾値違反モード)に設定すると、各計測項目の閾値に設定された値をカウント値がオーバーフローすると、ブレークが発生する。最小カウントはアンダーフローでブレークが発生。閾値違反モードで実行すると、ex\_rh\_timereadで参照しても計測結果は読み出せない。設定した閾値が読み出される。なお、type = EX\_TIM\_TYPENEW(最新カウント)のときは、閾値に設定された値をカウント値がオーバーフローしてもブレークは発生しない。なお、ブレークはe\_eventで設定したイベントでタイマ計測を終了したときに閾値違反となっていた場合に発生する。(カウント中にカウンタが閾値を超えた場合でも、すぐにブレークは発生しない。)ただし、e\_event=0を指定してブレークと同時に時間計測を停止する場合は、カウンタが閾値を超えたときにブレークが発生する。

開始イベント、終了イベントに設定できるイベントは、それぞれex\_rh\_evncondで設定したTEU\_CPU条件のイベント1個、または、ex\_rh\_evnintgcondで設定したTEU\_CPU条件のシーケンシャルイベント1個のみ。それ以外のイベントを設定した場合はエラー(EX\_PE\_INVEVNCOND)を返す。

ホットプラグインRUN中(接続～強制ブレークするまで)は、EX\_SE\_USRPGMRUNエラーとする。

(次開発バージョン以降の将来的な仕様として、ホットプラグインRUN中も通常RUN中と同様に参照可能とする)

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によって区間計測(DCUクロック利用)機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_DCKTIMENOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[13]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)に対して処理される。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定したタイマ番号を指定した場合はエラー(EX\_PE\_ANOTHERPETIM)が返る。

(4)開始/終了イベントに設定可能なイベントは現在選択されている物理CPU(PE)で設定されたイベント番号のみ。現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定されたイベント番号を指定した場合はエラー(EX\_PE\_ANOTHERPEEVN)が返る。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合、タイマ計測結果のクリアを行う。

(2)新規作成したTMU\_TCKタイマはデフォルトで「有効」状態になる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_timemode **[This function is not supported]** | SYNC | EMU | － | BRK | － | TRC | － | TIM | － | COV | － | HOT | － |
| ASYNC | EMU1 | － | EMU2 | － | BRK1 | － | BRK2 | － | TRC1 | － | TRC2 | － |

[Function]

タイマの動作モードの設定/参照を行う。

[Format]

int ex\_timemode( USHORT func, USHORT time\_no, USHORT \*mode, USHORT \*tidiv, USHORT \*ctcksel )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <--> |  | -- |
| USHORT | time\_no | <--> |  | -- |
| USHORT | \*mode | <--> |  | -- |
| USHORT | \*tidiv | <--> |  | -- |
| USHORT | \*ctcksel | <--> |  | -- |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | ----- | ----- |
|  |  |  |
| time\_no | | |
|  | ----- | ----- |
|  | | |
| \*mode | | |
|  | ----- | ----- |
|  |  |  |
| \*tidiv | | |
|  | ----- | ----- |
|  |  |  |
| \*ctcksel | | |
|  | ----- | ----- |
|  |  |  |

[Return Values]

EX\_NOSUPRT This function is not supported

[Description]

[デバッガ関係者向け注意事項]

(1)サポートしないため、常にEX\_NOSUPRTを返す。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_performancesw | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

パフォーマンス計測動作(有効/無効)の切り替えを行う。

[Format]

int ex\_rh\_performancesw( EXPFMSW \*sw )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| EXPFMSW | \*sw | <I > | 有効/無効情報 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | pfm\_sw | <I > | 動作の切り替え | ○ |
|  | USHORT | num | <I > | 切り替えるパフォーマンス数 | ○ |
|  | USHORT | \*pfm\_no | <I > | パフォーマンス番号(複数個可) | ○ |
| }EXPFMSW | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| pfm\_sw | | |
|  | EX\_PFM\_SW\_OFF | 無効 |
|  | EX\_PFM\_SW\_ON | 有効 |
|  | | |
| num | | |
|  | <1 - FFFFh> | 指定するパフォーマンス数 |
|  | | |
| \*pfm\_no | | |
|  | <0, 1 - FFFFh> | パフォーマンス番号 |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_ALREADY 既に同じ状態になっている

EX\_SE\_NONPFMNO パフォーマンス番号が存在しない

EX\_SE\_USRPGMRUN During execution of user program

EX\_PE\_ANOTHERPEPFM 別のPEのパフォーマンス番号は処理できない

EX\_NOSUPRT This function is not supported

EX\_PE\_PFMNOSPT パフォーマンス機能未サポート

[Description]

[Argument description]

(1)pfm\_sw

動作の切り替えを指定する。

EX\_PFM\_SW\_OFF ：無効(開始イベントが発生しても、計測されない)

EX\_PFM\_SW\_ON ：有効

(2)num

有効/無効を切り替えるパフォーマンス数を指定する。

(3)pfm\_no

有効/無効を切り替えるパフォーマンス番号を指定する。

numで指定したパフォーマンス数だけ、パフォーマンス番号を指定可能。

[動作説明]

パフォーマンス計測の有効/無効を切り替える。

無効状態では開始イベントが発生してもパフォーマンス計測されない。パフォーマンス計測を行うには対象のパフォーマンス番号の動作を有効にしておく必要がある。

ブレーク中のみ実行可能で、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)が返る。

存在しないパフォーマンス番号を指定した場合はエラー(EX\_SE\_NONPFMNO)が返る。

既に有効[無効]状態のパフォーマンス計測に対して、有効[無効]に設定しようとした場合はエラー(EX\_SE\_ALREADY)を返す。

num = 1, pfm\_no = 0ですべてのパフォーマンス番号を指定可能。同期デバッグモードの場合、全PEのパフォーマンス計測が操作対象となるが、非同期デバッグモードの場合は選択PEのパフォーマンス計測のみが操作対象となる。すべてのパフォーマンス計測を指定した場合、EX\_SE\_ALREADYのエラーは返らない。このエラーは無視してすべてのパフォーマンス計測を操作する。num = 1以外でpfm\_no = 0を指定した場合はエラー(EX\_PE\_DATA)が返る。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によってパフォーマンス計測機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_PFMNOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[15]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のパフォーマンス番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のパフォーマンス番号を指定した場合はエラー(EX\_PE\_ANOTHERPEPFM)が返る。ただし、num = 1, pfm\_no = 0で全パフォーマンス計測を指定した場合は、現在選択されている物理CPU(PE)以外の物理CPU(PE)も含むすべてのパフォーマンス計測を処理する。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合、パフォーマンス計測結果のクリアを行う。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_performanceclear | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

パフォーマンス計測結果の初期化を行う。

[Format]

int ex\_rh\_performanceclear( USHORT pfm\_no )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | pfm\_no | <I > | パフォーマンス番号 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| pfm\_no | | |
|  | <0, 1 - FFFFh> | パフォーマンス番号  0は全番号指定 |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONPFMNO パフォーマンス番号が存在しない

EX\_PE\_ANOTHERPEPFM 別のPEのパフォーマンス番号は処理できない

EX\_NOSUPRT This function is not supported

EX\_PE\_PFMNOSPT パフォーマンス機能未サポート

[Description]

[Argument description]

(1)pfm\_no

1. ：すべての設定済みパフォーマンス番号

1-FFFFh：個別のパフォーマンス番号を指定

[動作説明]

指定したパフォーマンス番号の計測結果を初期化する。

ブレーク中のみ実行可能で、ユーザプログラム実行中はエラー(EX\_SE\_USRPGMRUN)が返る。

存在しないパフォーマンス番号を指定した場合はエラー(EX\_SE\_NONPFMNO)が返る。

pfm\_no = 0ですべてのパフォーマンス番号を指定可能。同期デバッグモードの場合、全PEのパフォーマンス計測が操作対象となるが、非同期デバッグモードの場合は選択PEのパフォーマンス計測のみが操作対象となる。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によってパフォーマンス計測機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_PFMNOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[15]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のパフォーマンス番号に対して処理する。

(2)物理CPU(PE)の切り替えは、ex\_sounitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のパフォーマンス番号を指定した場合はエラー(EX\_PE\_ANOTHERPEPFM)が返る。ただし、num = 1, pfm\_no = 0で全パフォーマンス計測を指定した場合は、現在選択されている物理CPU(PE)以外の物理CPU(PE)も含むすべてのパフォーマンス計測結果を初期化する。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_performanceread | SYNC | EMU | × | BRK | ○ | TRC | ／ | TIM | ／ | COV | ／ | HOT | × |
| ASYNC | EMU1 | × | EMU2 | × | BRK1 | ○ | BRK2 | ○ | TRC1 | ／ | TRC2 | ○ |

[Function]

パフォーマンス計測結果の参照を行う。

[Format]

int ex\_rh\_performanceread( USHORT pfm\_no, EXPFMREAD \*pfm )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | pfm\_no | <I > | パフォーマンス番号 | ○ |
| EXPFMREAD | \*pfm | < O> | 計測結果 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | info | < O> | 計測結果情報 | ○ |
|  | USHORT | type | < O> | 計測項目 | ○ |
|  | ULONG | count | < O> | カウント | ○ |
| }EXPFMREAD | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| pfm\_no | | |
|  | <1 - FFFFh> | パフォーマンス番号 |
|  | | |
| info | | |
|  | ビット単位でオーバーフロー/閾値違反発生/計測中の情報を返す。 | |
|  | EX\_PFM\_OVFL | オーバーフロー |
|  | EX\_PFM\_TVFL | 閾値違反発生 |
|  | EX\_PFM\_COUNTINGNOW | 区間内で計測中 |
|  | | |
| type | | |
|  | 計測項目を返す。 | |
|  | ※計測項目の詳細は表 3‑23を参照。 | |
|  | | |
| count | | |
|  | <0 - FFFFFFFFh> | カウント値 |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONPFMNO パフォーマンス番号が設定されていない

EX\_PE\_ANOTHERPEPFM 別のPEのパフォーマンス番号は処理できない

EX\_NOSUPRT This function is not supported

EX\_PE\_PFMNOSPT パフォーマンス機能未サポート

[Description]

[Argument description]

(1)pfm\_no

パフォーマンス計測結果の参照を行うパフォーマンス番号を指定する。

ex\_rh\_performancesetで設定したパフォーマンス番号を指定すること。

(2)info

パフォーマンス計測結果に対するオーバーフロー/閾値違反発生/計測中の情報をビット単位で返す。

EX\_PFM\_OVFL ：オーバーフロー

EX\_PFM\_TVFL ：閾値違反発生

EX\_PFM\_COUNTINGNOW ：区間内で計測中

(3)type

計測項目が返る。

(4)count

計測結果が返る。

[動作説明]

pfm\_noで指定したパフォーマンス番号の計測結果を返す。

0や存在しないパフォーマンス番号を指定した場合は、エラー(EX\_SE\_NONPFMNO)を返す。

ユーザプログラムの実行中はエラー(EX\_SE\_USRPGMRUN)を返す。

計測した項目はtypeに返る。typeの詳細は表 3‑23を参照。

計測結果のオーバーフローや、閾値違反モードで計測した結果で閾値違反が発生した場合、イベント区間内で計測中だった場合はinfoに情報が返る。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によってパフォーマンス計測機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_PFMNOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[15]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)のパフォーマンス番号を処理する。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)のパフォーマンス番号を指定した場合はエラー(EX\_PE\_ANOTHERPEPFM)が返る。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_rh\_performanceset | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | × |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

パフォーマンス計測条件の設定/参照/削除を行う。

[Format]

int ex\_rh\_performanceset( USHORT func, USHORT pfm\_no, EXPFMINF \*pfm )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | func | <I > | 操作モード | ○ |
| USHORT | pfm\_no | <I > | パフォーマンス番号 | ○ |
| EXPFMINF | \*pfm | <IO> | 計測条件 | ○ |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| typedef struct | | |  |  |  |
| { | | |  |  |  |
|  | USHORT | mode | <IO> | 動作モード | ○ |
|  | USHORT | type | <IO> | 計測項目 | ○ |
|  | USHORT | s\_event | <IO> | 開始条件イベント番号 | ○ |
|  | USHORT | e\_event | <IO> | 終了条件イベント番号 | ○ |
|  | ULONG | count\_limit | <IO> | 閾値 | ○ |
|  | USHORT | htid | <IO> | ハードウェアスレッド指定 | ○ |
|  | USHORT | vmid | <IO> | 仮想マシン指定 | ○ |
| }EXPFMINF | | |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| func | | |
|  | EX\_SET | 設定(新規作成/変更) |
|  | EX\_REF | 参照 |
|  | ES\_DEL | 削除 |
|  | | |
| pfm\_no | | |
|  | <1 - FFFFh> | パフォーマンス番号 |
|  | | |
| mode | | |
|  | EX\_PFM\_ADD\_COUNT | 積算カウントモード |
|  | EX\_PFM\_MAX\_COUNT | 最大値保持モード |
|  | EX\_PFM\_MIN\_COUNT | 最小値保持モード |
|  | EX\_PFM\_NEW\_COUNT | 最新値保持モード |
|  | EX\_PFM\_PAS\_COUNT | パスカウントモード |
|  | EX\_PFM\_CNT\_LIMIT | 閾値違反モード(他のモードとORして使う) |
|  | | |
| type | | |
|  | 計測項目 | |
|  | ※詳細は表 3‑23を参照 | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| s\_event | | | |
|  | <0, 1 - FFFFh> | | 計測開始イベント番号  0は実行開始で計測も開始 |
|  | | | |
| e\_event | | | |
|  | <0, 1 - FFFFh> | | 計測終了イベント番号  0はブレークで計測も終了 |
|  | | | |
| count\_limit | | | |
|  | <0 - FFFFFFFFh> | | modeにEX\_PFM\_CNT\_LIMITをORしたときだけ有効 |
|  | | | |
| htid(G3世代専用。G4世代は0固定) | | | |
|  | bit0 | HT#0を計測する | |
|  | bit1 | HT#1を計測する (0固定とする) | |
|  | bit2 | HT#2を計測する (0固定とする) | |
|  | bit3 | HT#3を計測する (0固定とする) | |
|  | bit4 | HT#4を計測する (0固定とする) | |
|  | bit5 | HT#5を計測する (0固定とする) | |
|  | bit6 | HT#6を計測する (0固定とする) | |
|  | bit7 | HT#7を計測する (0固定とする) | |
|  | bit15-8 | 0固定 | |
|  |  | 設定の詳細は[Argument description]の<設定例>の表参照 | |
|  | | | |
| vmid(G3世代専用。G4世代は0固定) | | | |
|  | bit0 | VM#0を計測する | |
|  | bit1 | VM#1を計測する (0固定とする) | |
|  | bit2 | VM#2を計測する (0固定とする) | |
|  | bit3 | VM#3を計測する (0固定とする) | |
|  | bit4 | VM#4を計測する (0固定とする) | |
|  | bit5 | VM#5を計測する (0固定とする) | |
|  | bit6 | VM#6を計測する (0固定とする) | |
|  | bit7 | VM#7を計測する (0固定とする) | |
|  | bit15-8 | 0固定 | |
|  |  | 設定の詳細は[Argument description]の<設定例>の表参照 | |
|  | | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_FE\_COMM EXEC NOT STARTED

EX\_PE\_DATA Setting value is abnormal

EX\_SE\_USRPGMRUN During execution of user program

EX\_SE\_NONEVNNO イベント番号が存在しない(func = EX\_SET時)

EX\_SE\_NONPFMNO パフォーマンス番号が存在しない(func = EX\_DEL、EX\_REF時)

EX\_SE\_FULLPFMNO 全パフォーマンス計測が設定済み

EX\_PE\_INVEVNCOND 設定不可のイベントを指定した

EX\_PE\_ANOTHERPEEVN 別のPEのイベント番号は条件に設定できない

EX\_PE\_ANOTHERPEPFM 別のPEのパフォーマンス番号は処理できない

EX\_DE\_PFMNOMOUNT 指定したパフォーマンス計測はMCUに実装されていないため設定できない

EX\_NOSUPRT This function is not supported

EX\_SE\_CLKUSEDFORTIM CLKはタイマに使用されているためパフォーマンス計測に使用できない

EX\_PE\_PFMNOUSED パフォーマンス番号はタイマで使用済みのため使用できない

EX\_SE\_CANTCALL\_ASPRUNNING During execution of a user program using extended functions

EX\_SE\_SFTTRCRUN Executing user program with external soft trace single line output

EX\_PE\_PFMNOSPT パフォーマンス機能未サポート

[Description]

[Argument description]

(1)func

操作モードを指定する。

EX\_SET ：pfm\_noで指定したパフォーマンス番号を設定(新規作成/変更)

EX\_REF ：pfm\_noで指定したパフォーマンス番号の設定状態を参照

EX\_DEL ：pfm\_noで指定したパフォーマンス番号を削除

(2)pfm\_no

パフォーマンス番号を指定する。

0は指定不可。

(3)mode

動作モードを指定する。

EX\_PFM\_ADD\_COUNT ：積算カウントモード

EX\_PFM\_MAX\_COUNT ：最大値保持モード

EX\_PFM\_MIN\_COUNT ：最小値保持モード

EX\_PFM\_NEW\_COUNT ：最新値保持モード

EX\_PFM\_PAS\_COUNT ：パスカウントモード

EX\_PFM\_CNT\_LIMIT ：閾値違反モード

EX\_PFM\_CNT\_LIMITは他のmodeとORして使用する。

EX\_PFM\_PAS\_LIMITを指定した場合は(4)のtypeは無視してパスカウントを計測する。

(4)type

計測項目を指定する。

計測項目は表 3‑23に示すNoまたは定義で指定する。ただし、本開発バージョンではNo.53, 80, 81以外をサポートする。No53, 80, 81は使用しないこと。

表 3‑23 G3コアのパフォーマンス計測項目

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| カテゴリ | 分類 | No※1 | 定義 | 意味 |
| イベント発生回数 |  | 0 | EX\_PFM\_TYPE\_FETCHALL | すべての命令の実行回数 |
| 10 | EX\_PFM\_TYPE\_FETCHBRA | 分岐を伴う命令すべて |
| 25 | EX\_PFM\_TYPE\_EIINT | EIレベル割り込みの受付回数 |
| 26 | EX\_PFM\_TYPE\_FEINT | FEレベル割り込みの受付回数 |
| 27 | EX\_PFM\_TYPE\_INT | すべての命令非同期例外の受付回数 |
| 28 | EX\_PFM\_TYPE\_INTSYNC | すべての命令同期例外 |
| クロックサイクル数 | all | 34 | EX\_PFM\_TYPE\_CLKCYCL | クロックサイクル数 |
| 割り込み処理区間 | サイクル  カウント | 37 | EX\_PFM\_TYPE\_NOINTCYCL | 割り込み処理中ではない時間 |
| 38 | EX\_PFM\_TYPE\_DISABLEINT | DI/EIによる割り込み禁止区間 |
| フェッチ系 | 命令フェッチ | 50※2 | EX\_PFM\_TYPE\_IFUREQ | IFUが発行した命令フェッチリクエスト数 |
| 51※2 | EX\_PFM\_TYPE\_IFUHIT | 上記要求に対して命令Cacheにてノンウェイトでレスポンスした数 |
| 52※2 | EX\_PFM\_TYPE\_FROMREQ | FlashROMへのデータリクエスト数 |
| 53※2 | EX\_PFM\_TYPE\_VCIREQ | VCIバスへの命令フェッチリクエスト数 |
| FlashROMへの  データフェッチ | 80※2 ※3 | EX\_PFM\_TYPE\_FROMLDREQ | FlashROMへのデータリードリクエスト数 |
| 81※2 ※3 | EX\_PFM\_TYPE\_FROMLDHIT | 上記要求に対してデータSubCacheにてノンウェイトでレスポンスした数 |

※1 表に示すNo以外を指定した場合は、エラー(EX\_DE\_PFMNOMOUNT)を返す。

※2 コアがG3K(PCUを含む)/G3K-Hの場合は、キャッシュ機能を搭載しないためエラー(EX\_DE\_PFMNOMOUNT)を返す。

※3 コアがG3M-Hの場合は、エラー(EX\_DE\_PFMNOMOUNT)を返す。

G4コアの計測項目は表 3‑24に示すNoまたは定義で指定する。ただし、本開発バージョンではNo.52, 80以外をサポートする。No52, 80は使用しないこと

表 3‑24 G4コアのパフォーマンス計測項目

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| カテゴリ | 分類 | No※1 | 定義 | 意味 |
| イベント発生回数 |  | 0 | EX\_PFM\_TYPE\_FETCHALL | 命令実行数 |
| 10 | EX\_PFM\_TYPE\_FETCHBRA | 分岐処理を行った命令数(条件不一致のBcond/Loop命令と，例外命令は除く) |
| 11 | EX\_PFM\_TYPE\_BCOND | 条件分岐命令(Bcond/Loop)の実行数 |
| 12 | EX\_PFM\_TYPE\_BCONDMISS | 条件分岐命令(Bcond/Loop)の分岐予測ミス |
| 21※2 | EX\_PFM\_TYPE\_BGINT | バックグラウンド割り込み受付 |
| 22※2 | EX\_PFM\_TYPE\_BGEIINT | EIレベルバックグラウンド割り込み受付 |
| 23※2 | EX\_PFM\_TYPE\_BGFEINT | FEレベルバックグラウンド割り込み受付 |
| 24※2 | EX\_PFM\_TYPE\_BGTERMINT | 中断型例外(EIレベルとFEレベル含む)のバックグラウンド割り込み受付 |
| 25 | EX\_PFM\_TYPE\_EIINT | EIINTnの受付 |
| 26 | EX\_PFM\_TYPE\_FEINT | FEINTの受付 |
| 27 | EX\_PFM\_TYPE\_TERMINT | 中断型例外の受付(EIINTnとFEINTを含む) |
| 28 | EX\_PFM\_TYPE\_RESUMINT | 再実行型例外と完了型例外の受付 |
| クロック  サイクル数 | STALL | 29 | EX\_PFM\_TYPE\_STALLCYCL | 命令実行ユニットへの命令発行ストールサイクル |
| all | 34 | EX\_PFM\_TYPE\_CLKCYCL | クロックサイクル数 |
| 割り込み処理区間 | サイクル  カウント | 37 | EX\_PFM\_TYPE\_NOINTCYCL | 割り込み処理中では無いサイクル  (ISPRレジスタが0000hの期間) |
| 38 | EX\_PFM\_TYPE\_DISABLEINT | 割り込み処理中では無い，割り込み禁止サイクル(ISPRレジスタが0000hで，PSW.ID=1の期間) |
| フェッチ系 | 命令フェッチ | 50 | EX\_PFM\_TYPE\_IFUREQ | 命令フェッチのリクエスト |
| 51 | EX\_PFM\_TYPE\_IFUHIT | 命令キャッシュのヒット |
| 52 | EX\_PFM\_TYPE\_IFUBUSREQ | IFU AXI128への命令フェッチ・バスリクエスト |
| FlashROMへの  データフェッチ | 80 | EX\_PFM\_TYPE\_LSUBUSREQ | LSU AXI128へのデータ読み出しバスリクエスト |

※1 表に示すNo以外を指定した場合は、エラー(EX\_DE\_PFMNOMOUNT)を返す。

※2 G4MH V2.0以降で追加された項目のため、それより前のバージョンではエラー(EX\_DE\_PFMNOMOUNT)を返す。

(5)s\_event

計測区間の開始イベントに設定するイベント番号を指定する。

0を指定した場合は、実行と同時に計測を開始する。

(6)e\_event

計測区間の終了イベントに設定するイベント番号を指定する。

0を指定した場合は、ブレークと同時に計測を停止する。

(7)count\_limit

カウンタの閾値を設定する。

modeにEX\_PFM\_CNT\_LIMITをORしたときのみ有効。他のmodeでは設定を無視する。

(8)htid, vmid

計測対象のHT ID番号とVM IDフラグを設定する。

本パラメータはG3世代専用で、G4世代は0固定とする。

<設定例>

|  |  |  |  |
| --- | --- | --- | --- |
| 計測対象 | htid | vmid | 備考 |
| NMモード | 0000h | 0000h | 1VM/1HTではこの値のみ設定可能とする。  ※1はVMを計測する場合の設定値 |
| VM#0※1 | 0000h | 0001h |
| HT#0※1 | 0001h | 0000h |
| VM#0とVM#1 | 0000h | 0003h |  |
| VM#1とHT#1 | 0002h | 0002h | HTとVMの計測条件はORされる |

[動作説明]

イベント2点間のパフォーマンスを計測する。開始イベント番号に0を指定した場合は、実行と同時にパフォーマンス計測を開始し、終了イベント番号に0を指定した場合は、ブレークと同時に計測も停止する。ただし、開始条件や終了条件に0を指定した場合、本来RUN-BREAKタイマの計測に用いる指定のため、最大値、最小値、最新値は計測できなく、常に蓄積値が計測されるので注意。また、パスカウントは計測中に終了条件に一致した時点でカウントされるため、終了条件がない場合はカウントされないため注意。開始条件や終了条件に0を指定する使い方は推奨しない。

計測項目は、表 3‑23に示す。計測項目の自由度を上げるために、4ch個別に区間と計測項目を設定できる仕様になっている。

計測結果の読み出しはex\_rh\_performancereadで行う。

パフォーマンス計測条件は選択PEに対して設定される。対象のPEに切り替えてから計測条件を設定すること。

modeの指定を変えることで、設定区間での最大計測値/最小計測値/最新計測値/計測値の積算値を保持することができる。mode = EX\_PFM\_PAS\_COUNTを指定すると、設定区間を通過した回数を計測することができる。同一区間の任意の計測項目の計測値の積算値とパスカウントを別々のチャネルで計測することにより平均値を算出することも可能。さらに、modeにEX\_PFM\_CNT\_LIMITをORすることで、計測値が閾値違反するとブレークを発生させることも可能(閾値違反モード)。閾値違反モードで計測すると、ex\_rh\_performancereadで参照しても計測結果は読み出せない。設定した閾値が読み出される。なお、EX\_PFM\_CNT\_LIMITとEX\_PFM\_NEW\_COUNT(最新値保持モード)をORしたときは、計測値が閾値違反してもブレークは発生しない。また、ブレークはe\_eventで設定したイベントでパフォーマンス計測を終了したときに閾値違反となっていた場合に発生する。(カウント中にカウンタが閾値を超えた場合でも、すぐにブレークは発生しない。) ただし、e\_event=0を指定してブレークと同時に時間計測を停止する場合は、カウンタが閾値を超えたときにブレークが発生する。

開始イベント、終了イベントに設定できるイベントは、それぞれex\_rh\_evncondで設定したTEU\_CPU条件のイベント1個、または、ex\_rh\_evnintgcondで設定したTEU\_CPU条件のシーケンシャルイベント1個のみ。それ以外のイベントを設定した場合はエラー(EX\_PE\_INVEVNCOND)を返す。

パフォーマンス番号は共通で管理するため、PEごとに異なるパフォーマンス番号しか設定できない。別のPEで使用中のパフォーマンス番号を指定した場合、エラー(EX\_PE\_ANOTHERPEPFM)を返す。タイマ番号とパフォーマンス番号は共通で管理するため、タイマ番号と同じパフォーマンス番号も使用できない。使用した場合はエラー(EX\_PE\_PFMNOUSED)を返す。

TMU\_CLKはパフォーマンス計測機能以外にTMU\_CLKタイマでも同じリソースが使用される。このためTMU\_CLKタイマと排他使用となる。そのため、本関数を呼び出した時点でTMU\_CLKタイマの計測条件が設定済みの場合、エラー(EX\_SE\_CLKUSEDFORTIM)を返す。

E2拡張機能が有効に設定された状態で、ユーザプログラム実行中で呼び出された場合はエラー(EX\_SE\_CANTCALL\_ASPRUNNING)を返す。

E2拡張機能が無効に設定された状態で外部ソフトトレース単線出力機能が有効に設定されているときに、ユーザプログラムが実行中(LPDトレースモード中)で呼び出された場合は、エラー(EX\_SE\_SFTTRCRUN)を返す。外部ソフトトレース単線出力機能の詳細は、「外部ソフトトレース単線出力機能に関する動作仕様」参照。

RH850 G4世代MCUでは、MCUおよび物理CPU(PE)によってパフォーマンス計測機能が実装されていない場合がある。該当する場合に本I/Fを呼び出すとエラー(EX\_PE\_PFMNOSPT)を返す。

なお、機能の実装有無は、ex\_rh\_getenvのfuncinfo[15]を参照すること。

[マルチコア時の注意点]

(1)現在選択されている物理CPU(PE)に対して処理される。

(2)物理CPU(PE)の切り替えは、ex\_socunitinfo2で行う。

(3)現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定したパフォーマンス番号を指定した場合はエラー(EX\_PE\_ANOTHERPEPFM)が返る。

(4)開始/終了イベントに設定可能なイベントは現在選択されている物理CPU(PE)で設定されたイベント番号のみ。現在選択されている物理CPU(PE)とは別の物理CPU(PE)で設定されたイベント番号を指定した場合はエラー(EX\_PE\_ANOTHERPEEVN)が返る。

[デバッガ関係者向け注意事項]

(1)前回と異なるパラメータを設定した場合、パフォーマンス計測結果のクリアを行う。

(2)新規作成したパフォーマンス計測機能はデフォルトで「有効」状態になる。

(3)パフォーマンス計測機能はTMU\_CLKタイマと計測リソースが同じであるため排他で使用すること。TMU\_CLKタイマで計測条件を設定済みの場合はエラー(EX\_SE\_FULLPFMNO)を返す。

### ハードウェア直接操作用関数

ハードウェアを直接操作する関数。本関数でハードウェアのレジスタ等を書き換えたときの動作は保証できない。

個別ユーザ対応等で使う場合は、操作内容を十分検証した上で使うこと。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D\_Command | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

ホストI/F経由で直接アクセスできる資源(ICEレジスタ)のデータ読み出しを行う。

[Format]

int D\_Command( ULONG addr, USHORT len, UCHAR \*data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | addr | <I > | ICEレジスタアドレス | ○ |
| USHORT | len | <I > | バイト数 | ○ |
| UCHAR | \*data | < O> | データを保存するバッファ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| addr | | |
|  | <0 - FFFFFFFCh> | 4バイトバウンダリで設定のこと |
| len | | |
|  | <0 - FFFCh> | 4バイトバウンダリで設定のこと |
| \*data | | |
|  |  |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

[Description]

ホストI/F経由で直接アクセスできる資源(ICEレジスタ)のデータ読み出しを行う

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E\_Command | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

ホストI/F経由で直接アクセスできる資源(ICEレジスタ)のデータ変更を行う。

[Format]

int E\_Command( ULONG addr, USHORT len, UCHAR \*data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | addr | <I > | ICEレジスタアドレス | ○ |
| USHORT | len | <I > | バイト数 | ○ |
| UCHAR | \*data | <I > | 書き込みデータバッファ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| addr | | |
|  | <0 - FFFFFFFCh> | 4バイトバウンダリで設定のこと |
| len | | |
|  | <0 - FFFCh> | 4バイトバウンダリで設定のこと |
| \*data | | |
|  |  |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

[Description]

ホストI/F経由で直接アクセスできる資源(ICEレジスタ)のデータ変更を行う

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_dcuregread | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

ホストI/F経由で直接アクセスできるLPD通信で、DCUレジスタのデータリードを行う。

[Format]

int ex\_dcuregread( USHORT code, ULONG \*data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | code | <I > | IRアドレス | ○ |
| ULONG | \*data | < O> | データを保存するバッファ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| code | | |
|  | <0 - FFFFh> |  |
| \*data | | |
|  | <0 - FFFFFFFFh> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_SE\_STANDBY\_NOSPRT　　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

ホストI/F経由で直接アクセスできるLPD通信で、DCUレジスタのデータリードを行う。

[codeの指定方法]

・G3世代のcode指定方法

|  |  |
| --- | --- |
| bit15 | 0：bit13-11の指定は無効。カレントコアの DCUレジスタにアクセスする。  1：bit13-11の指定は有効。 |
| bit14 | Unused (0 fixed) |
| bit13-11 | マップ番号(0：LDU、1-6：PE番号※、7：ブロードキャスト)  ※PE1=1、PE2=2、… |
| bit10-8 | バンク番号 |
| bit7-0 | IRコード |

・G4世代のcode指定方法

|  |  |
| --- | --- |
| bit15 | 0：bit13-10の指定は無効。カレントコアのDCUレジスタにアクセスする。  1：bit13-10の指定は有効。 |
| bit14 | Unused (0 fixed) |
| bit13-10 | マップ番号(0：LDU、1-14：PE番号※、15：ブロードキャスト)  ※PE0=1、PE1=2、… |
| bit9-8 | バンク番号 |
| bit7-0 | IRコード |

(例)

DBG\_CTRLP(バンク番号:1、IRコード：0x0C)にアクセスする場合

[G3世代]

カレントコアのDBG\_CTRLPにアクセスする場合:code = 0x010C

PE2のDBG\_CTRLPにアクセスする場合:code = 0x910C

[G4世代]

カレントコアのDBG\_CTRLPにアクセスする場合:code = 0x010C

PE2のDBG\_CTRLPにアクセスする場合:code = 0x8D0C

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_dcuregwrite | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

ホストI/F経由で直接アクセスできるLPD通信で、DCUレジスタのデータ変更を行う。

[Format]

int ex\_dcuregwrite( USHORT code, ULONG data )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| USHORT | code | <I > | IRアドレス | ○ |
| ULONG | data | <I > | 書き込むデータ | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| code | | |
|  | <0 - FFFFh> |  |
| data | | |
|  | <0 - FFFFFFFFh> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_SE\_STANDBY\_NOSPRT　　During the applicable standby mode, commands cannot be executed because debugging operations cannot be performed. To continue debugging, perform an operation or a forced reset with the standby mode released.

[Description]

ホストI/F経由で直接アクセスできるLPD通信で、DCUレジスタのデータ変更を行う。

codeの指定方法はex\_dcuregreadを参照。

[初期停止・スタンバイモードデバッグ環境が有効時の注意事項]

(1)ユーザプログラムが実行中、初期停止とCyclicRunモードを除くスタンバイモード状態では、デバッグモジュールに対する制御が行えないため、エラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。ただし、ICUM有効の場合はCyclicRunモード時もエラー(EX\_SE\_STANDBY\_NOSPRT)を返して処理を行わない。

[デバッガ開発者向け注意事項]

(1)code(IRアドレス)の設定値(FFFFh)は特殊コードである。特殊コードに関する詳細は、「5. ログ情報の出力仕様」の章を参照のこと。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_setselid | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

　IE850Aのシリアル番号を設定

[Format]

int ex\_setselid( ULONG num, char \*number )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | num | <I > | 設定するシリアル番号のバイト数 | ○ |
| char | \*number | <I > | シリアル番号をアスキーコードで指定 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| num | | |
|  | <0 – 31> |  |
| \*number | | |
|  | <0 – XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_PE\_DATA Setting value is abnormal

EX\_PE\_IDSIZEOVER 文字数が許容範囲を超えている

[Description]

IE850Aにシリアル番号の設定を行う。

IE850A以外の場合、何もせずEX\_NOERRORを返す。

シリアル番号を書き換えた場合、エミュレータの電源を切り再起動する必要がある。

　　本I/FはRH850G4デバイスを対象としたEXECでのみ使用可能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_getselid | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

　E2エミュレータ/IE850Aのシリアル番号を参照

[Format]

int ex\_getselid( ULONG \*num, char \*number )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | \*num | < O> | 設定されているシリアル番号のバイト数 | ○ |
| char | \*number | < O> | 設定されているシリアル番号をアスキーコードで格納 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| \*num | | |
|  | <0 - 31> |  |
| \*number | | |
|  | <0 – XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_PE\_IDBLANK IDがブランク状態で設定されていない

[Description]

E2エミュレータ/IE850Aのシリアル番号を参照する。

E2エミュレータ/IE850A以外の場合、何もせずEX\_NOERRORを返す。

本I/FはRH850G4デバイスを対象としたEXECでのみ使用可能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_setnid | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

　E2エミュレータ/IE850Aのニックネームを設定

[Format]

int ex\_setnid( ULONG num, char \*number )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | num | <I > | 設定するニックネームのバイト数 | ○ |
| char | \*number | <I > | ニックネームを半角英数大文字小文字で指定 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| num | | |
|  | <0 – 200> |  |
| \*number | | |
|  | <0 – (200文字の半角英数大文字小文字の文字列)> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_PE\_DATA Setting value is abnormal

EX\_PE\_IDSIZEOVER 文字数が許容範囲を超えている

[Description]

E2エミュレータ/IE850Aのニックネームを設定する。

E2エミュレータ/IE850A以外の場合、何もせずEX\_NOERRORを返す。

本I/FはRH850G4デバイスを対象としたEXECでのみ使用可能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ex\_getnid | SYNC | EMU | ○ | BRK | ○ | TRC | ○ | TIM | ○ | COV | ／ | HOT | ○ |
| ASYNC | EMU1 | ○ | EMU2 | ○ | BRK1 | ○ | BRK2 | ○ | TRC1 | ○ | TRC2 | ○ |

[Function]

　E2エミュレータ/IE850Aのニックネームを参照

[Format]

int ex\_getnid( ULONG \*num, char \*number )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | \*num | < O> | 設定されているニックネームのバイト数 | ○ |
| char | \*number | < O> | 設定されているニックネームを半角英数大文字小文字で格納 | ○ |
|  |  |  |  |  |

[Arguments]

|  |  |  |
| --- | --- | --- |
| \*num | | |
|  | <0 – 200> |  |
| \*number | | |
|  | <0 – (200文字の半角英数大文字小文字の文字列)> |  |
|  | | |

[Return Values]

EX\_NOERROR Successful completion

EX\_PE\_IDBLANK IDがブランク状態で設定されていない

[Description]

E2エミュレータ/IE850Aのニックネームを参照する。

E2エミュレータ/IE850A以外の場合、何もせずEX\_NOERRORを返す。

本I/FはRH850G4デバイスを対象としたEXECでのみ使用可能

## エラーコード

本章では、エラーコードの一覧を記述する。

エラーの先頭5文字は、エラーの種類を示している。エラーコードの定義を、表 3‑25に示す。

共通のエラーコード一覧を表 3‑26に示す。

表 3‑25 エラーコードの定義

|  |  |  |
| --- | --- | --- |
| 先頭文字 | 意味 | 規定値の範囲 |
| EX\_ | Successful completion | 0x0000 |
| EX\_ | 本関数は未サポート | 0x0002 |
| EX\_FE | Fatal Error | 0x0100 - 0x01FF |
| EX\_UE | User system Error | 0x0200 - 0x02FF |
| EX\_SE | Status Error | 0x0300 - 0x03FF |
| EX\_PE | Parameter Error | 0x0400 - 0x04FF |
| EX\_DE | Device depend Error | 0x0500 - 0x05FF |
| EX\_EE | Exec I/F Error | 0x0600 - 0x06FF |
| EX\_QC | Custom Exec I/F Error | 0x0700 - 0x07FF |

表 3‑26 共通のエラーコード一覧

|  |  |  |  |
| --- | --- | --- | --- |
| 種類 | 名前 | 意味 | 対象 |
| Successful completion | EX\_NOERROR | 正常 | ○ |
| EX\_NOSUPRT | 本関数は未サポート | ○ |
| 致命的エラー | EX\_FE\_COMM | 通信がオープンされていない | ○ |
| EX\_FE\_NETFILE | ネットワーク情報ファイル | × |
| EX\_FE\_HOSTNAME | 指定されたホスト名に該当するホストがない | × |
| EX\_FE\_SENDOUT | IE-EXECへのデータ送信が時間内に終了しなかった | ○ |
| EX\_FE\_RECVOUT | IE-EXECの処理が時間内に終了しなかった | × |
| EX\_FE\_DEVFILE | デバイスファイルの読み込みに失敗 | ○ |
| EX\_FE\_DATACOMM | 受信したデータに異常があった | × |
| EX\_FE\_HANDLE | ハンドル値の異常 | × |
| EX\_FE\_NETDATA | ネットワーク情報ファイルに誤りがある | × |
| EX\_FE\_USBFAIL | USB通信が切れた(ソフトを終了させてICEの電源を入れなおす必要がある) | ○ |
| EX\_FE\_ANOTHEREXECRUN | ディレクトリが異なる別のEXECが既に動作している。USBドライバ競合の危険がある | ○ |
| EX\_FE\_MONTIMEOUT | モニタの処理が時間内に終了しなかった | ○ |
| EX\_FE\_EXECFILE | エグゼキュータファイルの読み込みに失敗 | × |
| EX\_FE\_BKNONCNCT | BKボードが接続されていない | × |
| EX\_FE\_EMNONCNCT | EMボードが接続されていない | × |
| EX\_FE\_IRGBOARDSET | ボード構成に矛盾がある | × |
| EX\_FE\_PODNONCNCT | POD/EM1ボードが接続されていない | × |
| EX\_FE\_EXECRUN | エグゼキュータ起動中 | ○ |
| EX\_FE\_MICROFILE | マイクロプログラムの読み込みに失敗 | × |
| EX\_FE\_INTFILE | INIファイルが見つからない | × |
| EX\_FE\_SENDBUFFOVER | 送信したデータに異常があった | × |
| EX\_FE\_RPRMNONCNCT | 端子エミュレータが搭載されていない | × |
| EX\_FE\_FLASHFILE | フラッシュF/Wファイルの読み込みに失敗 | × |
| EX\_FE\_IRGDEVFILE | デバイスファイルフォーマットタイプエラー | × |
| EX\_FE\_OLDDRIVER | デバイスドライバが旧バージョンである | × |
| EX\_FE\_INIDATA | INIファイルに誤りがある | × |
| EX\_FE\_MONITOR | ROMモニタのコマンド実行に失敗 | × |
| EX\_FE\_NOWINEXEC | シミュレータが起動できなかった | × |
| EX\_FE\_NOINITSIM | シミュレータ起動時のメモリが不足している | × |
| ユーザシステム異常 | EX\_UE\_VERIFY | ベリファイエラー | ○ |
| EX\_UE\_BUSHOLD | バスホールド | × |
| EX\_UE\_ENFBRK | 強制ブレークできない | ○ |
| EX\_UE\_RESET | リセット継続中 | ○ |
| EX\_UE\_MAPMEMFAIL | マッピング領域が大き過ぎて確保できなかった | × |
| EX\_UE\_COVMEMFAIL | カバレッジマッピング領域が大き過ぎて確保できなかった | × |

## レジスタID

＜レジスタIDの構成＞

EXEC I/Fでは、プログラムレジスタ、およびシステムレジスタはIDコードで構成される16bitデータを

使用する。

表 3‑27 レジスタID一覧

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| レジスタ分類 | | 名称 | ID | 名称 | ID | 名称 | ID | 名称 | ID |
| プログラムレジスタ | |  |  |  |  |  |  |  |  |
|  | 汎用レジスタ | r0 | 0000H | r1 | 0001H | r2 | 0002H | r3 | 0003H |
|  |  | r4 | 0004H | r5 | 0005H | r6 | 0006H | r7 | 0007H |
|  |  | r8 | 0008H | r9 | 0009H | r10 | 000AH | r11 | 000BH |
|  |  | r12 | 000CH | r13 | 000DH | r14 | 000EH | r15 | 000FH |
|  |  | r16 | 0010H | r17 | 0011H | r18 | 0012H | r19 | 0013H |
|  |  | r20 | 0014H | r21 | 0015H | r22 | 0016H | r23 | 0017H |
|  |  | r24 | 0018H | r25 | 0019H | r26 | 001AH | r27 | 001BH |
|  |  | r28 | 001CH | r29 | 001DH | r30 | 001EH | r31 | 001FH |
|  |  |  |  |  |  |  |  |  |  |
|  | プログラムカウンタ | PC | 0020H |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |
| ベクトルレジスタ | | vr0 /wr0 | 0030H | vr1 /wr1 | 0031H | vr2 /wr2 | 0032H | vr3 /wr3 | 0033H |
|  | | vr4 /wr4 | 0034H | vr5 /wr5 | 0035H | vr6 /wr6 | 0036H | vr7 /wr7 | 0037H |
|  | | vr8 /wr8 | 0038H | vr9 /wr9 | 0003H | vr10/wr10 | 003AH | vr11/wr11 | 003BH |
|  | | vr12/wr12 | 003CH | vr13/wr13 | 003DH | vr14/wr14 | 003EH | vr15/wr15 | 003FH |
|  | | vr16/wr16 | 0040H | vr17/wr17 | 0041H | vr18/wr18 | 0042H | vr19/wr19 | 0043H |
|  | | vr20/wr20 | 0044H | vr21/wr21 | 0045H | vr22/wr22 | 0046H | vr23/wr23 | 0047H |
|  | | vr24/wr24 | 0048H | vr25/wr25 | 0049H | vr26/wr26 | 004AH | vr27/wr27 | 004BH |
|  | | vr28/wr28 | 004CH | vr29/wr29 | 004DH | vr30/wr30 | 004EH | vr31/wr31 | 004FH |
|  | |  |  |  |  |  |  |  |  |

＜システムレジスタのレジスタID＞

システムレジスタは個別にレジスタIDを定義しない。

以下の算出式に従って算出したデータをレジスタ関連のEXEC関数に設定する必要がある。

サポートする必要があるselID、regIDについては、アーキテクチャ仕様、デバイス目標仕様を参照すること

算出式：0x0100 + (selID << 5) + regID

注：selID、regIDの範囲：0x00 - 0x1F

# 非同期通信インターフェース仕様

本章では、デバイスおよびEXEC内部のステータスの変化に応じて、その内容を非同期に上位層のソフトウェアに返すインターフェースを規定する。

## 概要

上位層のソフトウェアで関数の実態を準備したコールバック関数をEXEC内部で呼び出し、デバイスおよびEXEC内部のステータスの変化に応じてその内容を返す。

EXECでは、以下のタイミングでステータス変化が発生した場合に、コールバック関数を呼び出す。

・100msごとに発生させるWindowsタイマのコールバック処理でブレークステータスに変化があった場合  
デバイスステータスは400msecごとにステータスをチェックし変化があった場合  
仮想マシンステータスはブレーク中でかつ400msecごとにチェックし変化があった場合

・各EXEC関数コール時の前処理(EXEC関数多重チェック部)

・実行(ステップ実行含む)開始時

・ブレーク発生時

・トレース開始/停止時などの動作状態変化時

・エミュレータ接続時

なお、呼び出すためのコールバック関数のアドレス情報は、ex\_setcallback\_sts関数を発行して、EXECに渡す必要がある。

## インターフェース仕様

コールバック関数のインターフェース仕様を以下に示す。

以下に示すインターフェース/パラメータ仕様を元に、上位層のソフトウェアでコールバック関数の定義を行うこと。

[Format]

void callback\_sts( ULONG status\_type, ULONG status\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 型 | 引数識別名 | 属性 | 説明 | 対象 |
| ULONG | status\_type | <I > | デバイスステータス/ブレークステータス情報タイプ | ○ |
| ULONG | status\_info | <I > | デバイスステータス/ブレークステータス情報 | ○ |

[Arguments]

|  |  |  |
| --- | --- | --- |
| status\_type | | |
|  | デバイスステータス/ブレークステータス情報タイプ | |
|  | EX\_WM\_CPUSTAT | デバイスステータス情報 |
|  | EX\_WM\_BRKSTAT | ブレークステータス情報 |
|  | EX\_WM\_ERRSTAT | エラーステータス情報 |
|  | EX\_WM\_VMSTAT | 仮想マシンステータス情報(G4世代のみ) |
|  |  |  |
| status\_info | | |
|  | デバイスステータス/ブレークステータス情報 | |

[Description]

[Argument description]

(1)status\_type

コールバック関数を呼び出したステータス情報のタイプを格納する。

EX\_WM\_CPUSTAT：status\_infoはデバイスステータス情報

EX\_WM\_BRKSTAT：status\_infoはブレークステータス情報

EX\_WM\_ERRSTAT：status\_infoはエラーステータス情報

EX\_WM\_VMSTAT：status\_infoは仮想マシンステータス情報

(2)status\_info

status\_typeに示す情報を格納する。

＜status\_type=EX\_WM\_CPUSTAT指定時＞

現在のデバイスのステータス情報をビット単位で格納する。

bit0：電源状態を示す。

(0：電源ON状態 1：電源OFF状態)

bit4：STOPモードを示す。

(0：通常状態 1：STOPモード)

bit5：リセット状態を示す。

(0：通常状態 1：リセット状態)

bit6：HALTモードを示す。

(0：通常状態 1：HALTモード)

bit7：仮想化支援機能マシンモードを示す。

(0：ネイティブマシンモード 1：仮想マシンモード)

※G3世代のみ。G4世代は0固定。

bit8：ブレーク直前のマシンモードを示す。

(0：ネイティブマシンモード 1：仮想マシンモード)

※G3世代のみ。G4世代は0固定。

bit9：初期停止状態を示す。

(0：初期停止状態でない 1：初期停止状態)

bit10：DEEP STOPモードを示す。

(0：通常状態 1：DEEP STOPモード)

bit11：Cyclic RUNモード状態を示す。

(0：通常状態 1：Cyclic RUNモード状態)

bit12：Cyclic STOPモードを示す。

(0：通常状態 1：Cyclic STOPモード)

bit13：Cyclic RUNモードまたはCyclic STOPモードかつ無効状態を示す。

(0：通常状態 1：無効状態)

bit[31:14]：Unused (0 fixed)

＜status\_type=EX\_WM\_BRKSTAT指定時＞

ブレーク発生時のステータス情報をビット単位で格納する。

bit0：実行状態を示す。

(0：ブレーク状態 1：実行状態)

bit1：ステップ実行状態を示す。

(0：ブレーク状態 1：ステップ実行状態)

bit2：トレース状態を示す。

(0：停止状態 1：トレース実行状態)

bit3：仮想化支援機能有効時、デバッグ対象のCPU動作モードへの遷移で、タイムアウトの発生

を示す。（RH850 G4世代MCUのみ）

(0：なし 1：発生)

bit[7:4]：Unused (0 fixed)

bit[15:8]：ブレーク要因(デバッガ開発者向け注意事項の(3)参照)

bit[27:16]：Unused (0 fixed)

bit[31:28]：コア番号(非同期デバッグモード時のみ)

【RH850 G3世代MCUの場合】

0000：同期デバッグモード時は0固定

0001：PE1

0010：PE2

0011：PE3

0100：PE4

0101：PE5

0110：PE6

【RH850 G4世代MCUの場合】

0000：同期デバッグモード時は0固定

0001：PE0

0010：PE1

0011：PE2

0100：PE3

0101：PE4

0110：PE5

0111：PE6

1000：PE7

＜status\_type=EX\_WM\_ERRSTAT指定時＞

エラーが発生したときのエラー情報を格納する。発生したエラーを1とする。

bit0：Unused (0 fixed)

bit1：初期停止中のため同期ブレークできない

(EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILEDと同じ扱いのエラー)

bit2：IE850Aが冷却ファン停止を検出しました。ターゲット及びIE850Aの電源をOFFしてください

(EX\_FE\_FANERRORと同じ扱いのエラー)

bit3：IE850AがVOUTからの電源供給で過電流を検出しました

(EX\_FE\_VOUTERRORと同じ扱いのエラー)

bit4：IE850AがVOUTから電源供給を開始できませんでした

(EX\_FE\_BEFOREVOUTERRORと同じ扱いのエラー)

bit[31:5]：Unused (0 fixed)

Unused (0 fixed)

＜status\_type=EX\_WM\_VMSTAT指定時＞

選択しているコアのCPU動作モードを格納する。G4世代の仮想化支援機能サポート品種のみコールバックを返す。実行中は頻繁にCPU動作モードの切り替えが起こると想定されるため、一律で「実行中のため状態不明」を返す。

bit[3:0]：CPU動作モード

0000：ゲストモードでGPID=0

0001：ゲストモードでGPID=1

0010：ゲストモードでGPID=2

0011：ゲストモードでGPID=3

0100：ゲストモードでGPID=4

0101：ゲストモードでGPID=5

0110：ゲストモードでGPID=6

0111：ゲストモードでGPID=7

1000：ホストモード or 従来モード

1111：実行中のため状態不明

bit[31:4]：Unused (0 fixed)

[非同期デバッグモード時の注意事項]

(1)デバイスステータス情報は選択されているPEの情報のみを返す。

(2)ブレークステータス情報は全PEの情報を返す。

[デバッガ開発者向け注意事項]

(1)status\_type=EX\_WM\_BRKSTAT時、bit[1:0]のいずれかが1の場合、ブレーク要因は固定値(0)を格納する。

(2)現在選択されている物理CPU(PE)の情報を返す。ただし、非同期デバッグモード時は選択されていない物理CPU(PE)のブレークステータス情報も返す。

(3)ブレーク要因を以下に示す。なお、同時に発生した場合は優先順位の高い要因を格納する。イベント番号などの詳細情報については、ex\_getstatusおよびex\_rh\_hwbrkeventを個別に発行して取得する。

表 4‑1 ブレーク要因判定の優先順位

|  |  |  |
| --- | --- | --- |
| 優先順位 | ブレーク要因 | 内容 |
| 高 | EX\_PWOFF\_BK | ターゲット電源OFFによるブレーク |
| ↑ | EX\_SOFT\_BK | ソフトウェアブレーク |
| ｜ | EX\_ENF\_BK | 強制ブレーク |
| ｜ | EX\_TEMP\_BK | テンポラリブレーク |
| ｜ | EX\_EVENT\_BK | イベントブレーク |
| ｜ | EX\_TRCFULL\_BK | トレースフルブレーク |
| ｜ | EX\_TRCDLY\_BK | トレースディレイブレーク |
| ｜ | EX\_TIMTOTOVF\_BK | TMU\_CLKタイマ積算カウント閾値違反ブレーク |
| ｜ | EX\_TIMMAXOVF\_BK | TMU\_CLKタイマ最大カウント閾値違反ブレーク |
| ｜ | EX\_TIMMINUDF\_BK | TMU\_CLKタイマ最小カウント閾値違反ブレーク |
| ｜ | EX\_TIMCNTOVF\_BK | TMU\_CLKタイマパスカウント閾値違反ブレーク |
| ｜ | EX\_TIMTCKTVFL\_BK | TMU\_TCKタイマカウント閾値違反ブレーク |
| ｜ | EX\_PFMCNTTVFL\_BK | パフォーマンスカウント閾値違反ブレーク |
| ｜ | EX\_EXPFUNCACT\_BK | 拡張機能のアクションブレーク |
| ｜ | EX\_EXPFUNCFULL\_BK | 拡張機能のフルブレーク |
| ↓ | EX\_SFTTRCFULL\_BK | 外部ソフトトレース単線出力フルブレーク |
| 低 | EX\_RELAY\_BK | リレーブレーク |

(4)上位層のソフトウェアで実装するコールバック関数内部からEXEC I/Fを発行することはできない。

(5)status\_type=EX\_WM\_BRKSTAT時、ステップ後のブレークの応答では、ブレーク要因に固定値(0)を格納する。

(6)ブレークステータスは100msecごとに変化をチェックするが、デバイスステータスは頻繁にステータスが変わる場合があるため400msecごとにチェックするため、400msec以下のステータス変動は検出できない場合がある。

(7)IE850Aの電源供給エラーおよび冷却ファンエラーに関して、デバッグ開始時にエラーが発生した場合はデバッグ開始できないようにすること。デバッグ中にエラーが発生した場合はエラーを返すのみ行うこと。デバッグ中にエラーが発生した場合にデバッガが自動でデバッグを終了し切断しないこと。

(8)非同期通信インターフェースを上位層で定義および使用する場合の実装例を以下に示す。

[実装例]

[インクルードするex\_v850.hの記述内容]

typedef void(\* EXCALLBACK)(ULONG, ULONG);

\_API\_TYPE\_ ex\_setcallback\_sts(EXCALLBACK);

[上位層ソフトウェアでの実装例]

#include "ex\_v850.h"

void main( void )

{

EXCALLBACK ptFunc = callback\_sts;

// EXECへコールバック関数ポインタを設定

ex\_setcallback\_sts( ptFunc );

:

}

void callback\_sts( ULONG status\_type, ULONG status\_info )

{

if( status\_type == EX\_WM\_CPUSTAT )

{

// EXECから送信された「現在のデバイスのステータス情報(status\_info)」を取得

// [注意]取得情報は、現在EXECに設定中のPEの情報

// 実際の取得処理を記述 //

}

else if( status\_type == EX\_WM\_BRKSTAT )

{

// EXECから送信された「ブレーク発生時のステータス情報(status\_info)」を取得

// [注意]取得情報は、現在EXECに設定中のPEの情報

// 実際の取得処理を記述 //

}

}

(9)status\_type=EX\_WM\_ERRSTAT指定時で情報取得する場合は、以下の対応を行う。

1. bit[7:0]に定義しているエラー情報を取得する。(ユーザへエラーを通知する)
2. ①で取得したエラーが「初期停止中のため同期ブレークできない」(EX\_SE\_FETCHSTOP\_AFTER\_BRKFAILED)の場合は、継続してデバッグを進めることができないため、ex\_resetを入れるようにユーザにガイドする

(10)初期停止・スタンバイモードデバッグ環境が有効時は、ユーザプログラム停止中においても、CPUステータスが変化する可能性がある。

(例：レジスタライトにより初期停止解除)

このため、ユーザプログラム停止中でもex\_getstatusまたは非同期通信インターフェースのコールバック関数を用いたCPUステータスの監視を行うこと。

# ログ情報の出力仕様

本章では、各EXEC I/Fの処理結果の情報を外部ファイル(以下、ログファイル)に出力する仕様について規定する。

## 概要

EXECでは、各EXEC I/Fの処理結果(以下、ログ情報)をログファイルに出力する機能(以下、ログ機能)を実装する。

ログファイルに出力するログ情報を以下に示す。なお、出力するログ情報は、バイナリ形式でログファイルに書き込む。

[出力するログ情報]

(1)発行されたEXEC I/F名

(2)戻り値情報

(3)パラメータ情報(Input/Outputすべて。ただしセキュリティ関連のパラメータは出力しない。)

ログ機能に関する情報は、EXEC起動時に読み込むINIファイルに記載する。

また、EXEC起動後でも、ログファイルへのログ情報出力を切り替えられるように、既存のI/Fを使用して実装する。

INIファイルからのログ情報の取得タイミング、ログファイルを開くタイミングおよびログファイルを閉じるタイミングは以下のようになる。

[INIファイルからのログ情報取得タイミング]

(1)デバッガがEXEC DLLをロードしたタイミング

[ログファイルを開くタイミング]

(1)デバッガがEXEC DLLをロード後、最初に発行されるEXEC I/F関数

(2)ex\_exit関数を発行後、最初に発行されるEXEC I/F

[注意事項]

(1)INIファイルに指定したフォルダ内に、指定したファイル名のファイルが存在しない場合のみログファイルを作成してログ情報を書き込む。

(2)同一名のファイルがある場合は、ログファイルを開かずログ情報を出力しない。この状態になると、EXEC DLLを一度アンロード(および同一名ファイルの削除/移動)しない限り、ログ情報を出力しない。

[ログファイルを閉じるタイミング]

(1)デバッガがEXEC DLLをアンロードしたタイミング

(2)ex\_exit関数を発行したタイミング

## INIファイル仕様

INIファイルに記載するログ機能に関する情報を示す。

なお、INIファイルの[850.EX]セクションに、以下に示す3種類のKEYを使用して記載する。

(1)ログ機能自体の有効/無効情報

(2)ログ情報を出力するファイル名情報

(3)各機能単位でのログ情報出力の有効/無効情報およびEXEC起動時のログ情報出力の有効/無効情報

表 5‑1 INIファイル記載仕様

|  |  |  |  |
| --- | --- | --- | --- |
| No | INIファイル  KEY名 | 説明 | 記載例 |
| 1 | LogFuncEnable | ログ機能自体(以下に示す(2)(3)の情報)の有効/無効の情報を0/1で指定する。  0：ログ機能のすべてを無効にする。EXEC起動後のログファイル出力の切り替え機能も無効となる。  1：ログ機能のすべてを有効にする。EXEC起動後のログファイル出力の切り替え機能も有効となる。 | LogFuncEnable=1 |
| 2 | LogFileName | ログ情報を出力するログファイル名をフルパスで指定する。  標準Cライブラリのstrftimeの書式も使用可能  DLLがロードされた時刻が書式に従って展開される  使用できる書式はstrftimeの仕様を参照のこと  例.  LogFileName=exec-%y-%m-%d-%H%M.logと指定すると  exec-17-03-24-1529.logのように展開される。 | LogFileName=C:\Temp\execLog.log |
| 3 | LogOutEnable | 関連機能ごとにグループ化したEXEC I/Fのログ情報出力の有効/無効情報、およびEXEC起動時のログ情報出力の有効/無効情報をビット単位(16進数4byte)で指定する。  (各ビットの詳細は、表 5‑2を参照) | LogOutEnable=0007 |

表 5‑2 ログ情報出力の有効/無効情報(LogOutEnable)の詳細

|  |  |  |  |
| --- | --- | --- | --- |
| No | bit位置 | 説明 | 備考 |
| 1 | bit31 | 制限が有効の場合は、1つのパラメータに対して、最大で0x0100bytesまでログ出力を行い、それ以上は出力しない。無効の場合は、すべてログ出力を行う。 | 0：有効  1：無効 |
| 2 | bit[30:15] | 未使用。  EXEC内部では無視するが、0を指定すること。 |  |
| 3 | bit14 | 外部ソフトトレース単線出力関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 4 | bit13 | 非同期通信関連のEXEC I/F、コールバック関数のログ情報出力の有効/無効を指定する。(Phase1では未サポートとする。) |  |
| 5 | bit12 | ハードウェア直接操作関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 6 | bit11 | タイマ関連とパフォーマンス計測関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 7 | bit10 | カバレッジ関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 8 | bit9 | リアルタイムRAMモニタ関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 9 | bit8 | トレース関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 10 | bit7 | 実行関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 11 | bit6 | イベント条件関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 12 | bit5 | レジスタ操作関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 13 | bit4 | メモリ操作関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 14 | bit3 | ハードウェア条件設定-環境設定関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 15 | bit2 | ハードウェア条件設定-状態検出関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 16 | bit1 | 初期化関数(起動/終了/リセット)関連のEXEC I/Fのログ情報出力の有効/無効を指定する。 |  |
| 17 | bit0 | EXEC起動時点でのログ情報出力の有効/無効を指定する。 |  |

【注意】各ビットで指定する値の意味は以下の通り。(bit31を除く)

0：該当するビットの機能が無効

1：該当するビットの機能が有効

## ログ情報の出力切り替え用EXEC I/F仕様

EXEC起動後に、ログ情報の出力切り替えをサポートするために以下に示すEXEC I/Fを使用する。

[Format] int ex\_dcuregwrite( USHORT code, ULOG data );

[Arg] USHORT code：ログ情報の出力切り替え機能への遷移指定

ULONG data：ログ情報の出力の切り替え指定

[Description]

[Argument description]

(1)code

ログ情報の出力切り替え機能の処理へ遷移させるための特殊コードを指定する。

特殊コード：FFFFh固定。その他の値を指定した場合は無効。

(2)data

ログ情報の出力の切り替え情報をbit0に指定する。

なお、codeにFFFFhを指定した場合のみ有効。

0：ログ情報のログファイルへの書き込みを停止する。

1：ログ情報のログファイルへの書き込みを開始する。

[動作説明]

EXEC起動後に、ログ情報のログファイルへの出力を切り替えたい場合に使用する。

本関数は通常、上位層が直接DCUレジスタの値を変更するために使用するI/Fであるが、codeに特殊コード(FFFFh)を指定した場合のみ、ログ情報のログファイルへの出力切り替え処理に遷移して、dataで指定した値に出力の状態を切り替える動作を行う。

codeに特殊コード(FFFFh)以外を指定した場合は、通常のDCUレジスタの変更処理を行う。

物理CPU(PE)の選択に関係なく動作する。

## ログ情報フォーマット仕様

ログファイルに出力するログ情報のフォーマットを示す。

[EXEC発行番号][^][EXEC I/F識別子][エラーコード][^FF][^][パラメータ番号][パラメータ値][^FF]

⑨

⑧

⑦

⑥

⑤

④

③

②

①

⑩

①：EXEC I/Fの発行順序の番号(2byte)

②：区切り文字(1byte)

③：EXEC I/Fを識別するための番号ですべてのEXEC I/Fで一意の番号(1byte)

④：EXEC I/Fのエラーコード(4byte)

⑤：区切り文字(2byte)

⑥：区切り文字(1byte)

⑦：すべてのEXEC I/Fのすべてのパラメータで一意の番号(2byte)

⑧：実際のパラメータデータ(パラメータによる)

⑨：パラメータ情報の区切り文字(2byte)

⑩：各EXEC I/Fのパラメータの分だけ繰り返す。

【改定履歴】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版  数 | 作成日 | 変更内容 | 作成部門: | T開2EM1 | |
| 作成 | 審査 | 承認 |
| 01 | 2014/1/15 | ■P.39 ex\_iniexec  [Return Values]にEX\_FE\_TGTNONCNCTを追加 | 島田 | - | - |
| ↓ | 2014/2/19 | ■ex\_socunitinfo2(EX\_UNIT\_COREPESEL, EX\_UNIT\_SLAVEIDSEL)  　P.102, P109 [デバッガ開発者向け注意事項]の表中の備考欄を修正  ■(1)トレース  P.174　表中のNo.3、No.8についての備考欄を修正  ■ex\_trcsw, ex\_trcrun, ex\_trcmode, ex\_trcevent, ex\_rh\_trceventsel2, ex\_rh\_trcbufcont, ex\_rh\_trctune  　P.178, P.180, P.183, P.188, P.200, P.204, P.222  [マルチコアおよびスレーブ対応時の注意点]の内容を修正  ■ex\_trcsw, ex\_trcrun, ex\_trcmode, ex\_trcclear, ex\_trcevent, ex\_trcdelay, ex\_rh\_trceventsel2, ex\_rh\_trcbufcont, ex\_trcrawmeminfo, ex\_trcrawmemread, ex\_rh\_trchwevncopeinfo, ex\_rh\_trctune  　P.177, P.179, P.182, P.185, P.187, P.191, P.198, P.203, P.205, P.207, P.210, P217  　[Return Values]からEX\_NOSUPRTを削除 | 横山 | - | - |
| ↓ | 2014/2/21 | ■ex\_trcsw  　P.177 [Argument description]を修正 | 横山 | - | - |
| ↓ | 2014/2/25 | ■ex\_rh\_evncond  　P.150 [表3-5-3：デバイス別に搭載するTEUスレーブイベント資源]のPFC1Aに対してTEU\_GRMは、○～△に修正 | ビナイ | - | - |
| ↓ | 2014/3/4 | (3)タイマ関数/パフォーマンス計測関数  P.235 [三層対応時の注意点]を追加。  ■ex\_timesw  　P.236 [本関数は未サポート]の記述を削除  ■ex\_timeclear  　P.239 [本関数は未サポート]の記述を削除  ■ex\_rh\_timeread  　P.244 [本関数は未サポート]の記述を削除  ■ex\_timeevent2  　P.246 [本関数は未サポート]の記述を削除  ■ex\_rh\_timeevent  　P.252 [本関数は未サポート]の記述を削除  ■ex\_rh\_performancesw  　P.256 [本関数は未サポート]の記述を削除  ■ex\_rh\_performanceclear  　P.258 [本関数は未サポート]の記述を削除  ■ex\_rh\_performanceread  　P.259 [本関数は未サポート]の記述を削除  ■ex\_rh\_performanceset  　P.261 [本関数は未サポート]の記述を削除  　　　　[Format]を修正  　P.262 [Arguments]を修正  　P.263 [Description]-[Argument description]を修正 | 横山 | - | - |
| ↓ | 2014/3/12 | ■ex\_rh\_evncond  　P.146/P.151 [TEU仕様拡張により引数acc\_srcのbit指定の意味を追記]  　また、ビット指定の妥当チェックは行わないため、エラー記述について内容を修正。 | ビナイ | - | - |
| ↓ | 2014/4/2 | ■2.7 非同期デバッグに関する動作仕様  　P.32-33 新規追加  　これに伴い以下の各I/Fにエラーコードだけ追加  　　P.74 ex\_setenv P.191 ex\_trcdelay  　　P.120 ex\_memwrite P.198 ex\_rh\_trceventsel2  　　P.177 ex\_trcsw P.203 ex\_rh\_trcbufcont  　　P.179 ex\_trcrun P.205 ex\_trcrawmeminfo  　　P.182 ex\_trcmode P.207 ex\_trcrawmemread  　　P.185 ex\_trcclear P.210 ex\_rh\_trchwevncopinfo  　　P.187 ex\_trcevent P.217 ex\_rh\_trctune  ■ex\_getstatus  　P.62 「非同期デバッグモード時の注意点」を新規追加  ■ex\_socunitinfo2  　P.82,90-91,109,110 「EX\_UNIT\_ASYNCPEOPT」に関する記述を新規追加  　P.91 使用していないエラーコードを削除  ■ex\_memwrite  　P.120 使用していないエラーコードを削除  ■ex\_swbrkcond  　P.168-169 非同期デバッグに関する記述を新規追加  ■ex\_trcmode  　P.183 フルブレークモード/ディレイブレークモードと非同期デバッグモードの関係について説明を新規追加  ■4.非同期インターフェース仕様  　P.274-275 非同期デバッグに関する記述を新規追加 | 永井 | - | - |
| ↓ | 2014/4/2 | ■4.非同期インターフェース仕様  　P.273-276 デバイスステータスのチェックタイミングを変更 | 永井 | - | - |
| ↓ | 2014/4/3 | ■ex\_socunitinfo2  　P.97 オプションバイトのマスクパターン追加  　追加したシリーズ：R1L、P1M、C1M、C1H、F1H | 永井 | - | - |
| ↓ | 2014/4/8 | * P.69 ex\_boardinfo2   ENVファイル用の下記エラー定義を追加。  EX\_FE\_ENVFILE　　　 ENVファイルが見つからない。  EX\_FE\_ENVDATA ENVファイルに誤りがある。  ■P.72 ex\_boardinfo2  ENVファイル用のエラー定義の説明を追加。 | 畑原 | - | - |
| ↓ | 2014/4/10 | ■ex\_memwrite  　P.124-125 データフラッシュの書き込みサイズを1ブロック単位から4byte単位に変更 | 永井 | - | - |
| ↓ | 2014/4/10 | ■1.4.3 動作環境についての注意点  　P.19 台数制限数を変更、カウント対象について記載  ■ex\_initexec  　P.41 台数制限を超えた状態のエラー(EX\_FE\_ICENUMOVER)を追加  ■ex\_setemulator  　P.44-46  　・台数制限を超えてエミュレータが接続されている場合のstatを追加  　・ニックネーム機能を追加  　　ニックネーム指定方法と仕様上の注意点を記載  ■ex\_boardinfo2  　P.70 台数制限を超えた状態のエラー(EX\_FE\_ICENUMOVER)を追加 | 永井 | - | - |
| ↓ | 2014/4/17 | ■ex\_trcevent  　P.188,189 エラーを追加  　P.189 [デバッガ開発者向け注意事項]に注意事項を追加 | 横山 | - | - |
| ↓ | 2014/4/18 | ■3.2.7 アナライザ (1)トレース関数  　P.177 [スレーブトレースの注意事項]を追加 | 横山 | - | - |
| ↓ | 2014/4/22 | ■2.2.1 起動  P.27 ホットプラグイン接続による起動手順を追加  ■2.3 CPUリセットが発生する関数  P.29 ホットプラグイン接続時はCPUリセットが入らない旨を追加  ■3.1 記述規定  P.34 StatusにホットプラグインRUN中状態を追加  全I/FのStatusにホットプラグインRUN中の呼出し可否を追加  ■3.2 関数詳細説明  　ex\_setenv,ex\_memread,ex\_memwrite,ex\_memfill,ex\_regread,ex\_regwrite,ex\_rh\_evncond,ex\_rh\_evnintgcond,ex\_hwbrkevent,ex\_swbrkcond,ex\_trcrun,ex\_trcmode,ex\_trcclear,ex\_trcevent,ex\_trcdelay,ex\_rh\_trceventsel2,ex\_rh\_trcbufcont,ex\_trcrawmeminfo,ex\_trcrawmemread,ex\_rh\_trchwevncopeinfo,ex\_timeevent2,ex\_rh\_timeevent,ex\_pincont,ex\_socunitinfo2  　[動作説明]にホットプラグインRUN中の動作を追記。 | 島田 | - | - |
| ↓ | 2014/4/24 | ■Phase2.2向け修正 P9 表1-1.サポート機能一覧 Phase2.2用に更新  P22 表1-10 ファイルのプロパティ Phase2.2用に更新  P23 表1-11 ファイルセット一覧 ENVファイルを追加 | 石井 | - | - |
| ↓ | 2014/5/9 | ■ex\_rh\_timeevnet  　P.256 [動作説明] 閾値違反モードについての説明を追加  ■ex\_rh\_performanceset  　P.267 [動作説明] 閾値違反モードについての説明を追加 | 横山 | - | - |
| ↓ | 2014/5/9 | タイマ関数／パフォーマンス計測関数  　P.237 [デバッガ関係者向け注意事項]を追加 | 横山 | - | - |
| ↓ | 2014/5/9 | 表1-1.サポート機能一覧  サポートマイコン　CC-Cube　WSのみのサポートに変更  マルチコア対応　フェッチ停止エミュレーションを追加  接続・切断　シリアル番号による接続を追加  接続・切断　ニックネームによる接続を追加  ダウンロード・ アップロード機能　Data Flash　Phase2.2での対応について備考を追記 | 石井 | - | - |
| ↓ | 2014/5/12 | ■2.7 非同期デバッグに関する動作仕様  　P.34 表2-3 タイマ・パフォーマンス関連のI/Fを追加  ■タイマ・パフォーマンスの全操作時の動作について説明を追加  　P.241 ex\_timesw  　P.243 ex\_timeclear  　P.262 ex\_rh\_performancesw  　P.263 ex\_rh\_performanceclear | 永井 | - | - |
| ↓ | 2014/5/20 | ■3.2 関数詳細説明  　[Arguments] と[動作説明]にホットプラグイン接続指定を追記。  　ex\_initexec  　[デバッガ関係者向け注意事項]のホットプラグインRUN中における  不要な記述を削除  ex\_regread,ex\_regwrite | 島田 | - | - |
| ↓ | 2014/5/23 | ■ex\_timeevent2  　P.252 [動作説明] 閾値違反モードについての説明を追加  ■ex\_rh\_timeevnet  　P.258 [動作説明] 閾値違反モードについての説明を追加  ■ex\_rh\_performanceset  　P.269 [動作説明] 閾値違反モードについての説明を追加 | 横山 | - | - |
| ↓ | 2014/5/28 | ■ex\_rh\_trceventsel2  　P.201  　[Return Values]からEX\_NOSUPRTを削除(Rev.2.23の修正漏れ)  ■ex\_rh\_performanceset  　P.267 [Arguments] – htid, vmidの説明を修正 | 横山 | - | - |
| ↓ | 2014/5/28 | ■ex\_socunitinfo2  P.113 unit\_type別 EXECの状態と関数の呼出し可/不可の関係の備考を訂正 | 島田 | - | - |
| ↓ | 2014/5/28 | ■ex\_rh\_trctune  　P.216 [Arguments] – trc\_tunetype=EX\_TRC\_CPUSYNCMSG の説明を修正  　P.221 [Description] – [動作説明] - <trc\_tunetype=EX\_TRC\_CPUSYNCMSG>の時 の説明を修正 | 横山 | - | - |
| ↓ | 2014/5/30 | ■ex\_setemulator  　P.46 [Arguments] flag 「bit1-31」を「bit1-30」に修正  ■ex\_socunitinfo2  　P.94、112 「EX\_SE\_ASYNCMODE\_CANTSEL」を「EX\_SE\_ASYNCMODE」に修正 | 永井 | - | - |
| ↓ | 2014/6/9 | AXIクロック停止時にメモリアクセス不可の領域へのアクセス時のエラーコード(EX\_SE\_AXICLKDSB\_ACCFAILED)の定義を追加  　P.116 [Return Values]　ex\_memread  　P.123 [Return Values]　ex\_memwrite  　P.130 [Return Values]　ex\_memfill  　P.133 [Return Values]　ex\_memcopy | ビナイ | - | - |
| ↓  FD  FFD12014/07/2817:23:11佐々  　木宏  '14. 7.28  SPF-RH850G3EXEC(IF)\_k02\_k023173FFDD7FDE7FEBFFDB7FDCFFEB7FD07FCA7FE7FFE6FFD07FCA7FE7FFE6FFE67FE77FE43926BF53FFF93F5FB493B95C7FF97FEC7FE77FE5FFE8FFEFFFE47FE8FFE6FFE3FFF97FD67FD7FFDCFFE97FD6FFDBFFE3FFE57FE7FFDC7FE67FDD7  **佐々**  **'14. 7.28**  **木宏**  FFD22014/07/2411:30:02石  　井沢  '14. 7.24  SPF-RH850G3EXEC(IF)\_k02\_k01291FF39BCC7FFC9FAFDDC6DB837FFCBFF63FF3BFF2FFF47FF7FFF23FF47FF37FF2FFFCBFEB3FEBFFEE7FF4BFEB7FEDFFF1FFF2BFF3FFEE3FF33FEEBFE9FFEEBFEF3FF5FFEDBFEE7FF5BFE83FE53FF3FFF37FE83FE53FF3FFF3BFF37FF1B  **石**  **'14. 7.24**  **井沢** | 2014/6/9 | ■P12, P13 タイマ・パフォーマンス関連I/FをサポートI/Fに修正  ■P237, P245 タイマについての制限事項を削除 | 横山 | - | - |
| ↓ | 2014/6/11 | ■表1-1.サポート機能一覧(2/2)の補足変更  ○：サポート予定(Phase2.0以前の記載ではサポート済みを意味する)。  ↓  ○：サポート予定(Phase2.1以前の記載ではサポート済みを意味する)。 | 石井 | - | - |
| ↓ | 2014/6/13 | P.282 5.ログ情報の出力仕様 – 5.1概要 [出力するログ情報]の(3)パラメータ情報について修正 | 横山 | 石井  2014/6/16 | 佐々木  2014/6/16 |
| ↓ | 2014/6/23 | P.137-140  ex\_regread、ex\_regwrite  EX\_M\_DMM\_REG\_READおよびEX\_DMM\_REG\_WRITEのモードをサポートしない旨を記載。  また、これに合わせてP.12のEXEC仕様関連表も修正。 | 永井 | - | - |
| ↓ | 2014/6/23 | ■ex\_trcrawmeminfo  　P.208 [Format]-dbg\_mbaの説明を修正 | 横山 | 石井  2014/6/24 | 佐々木  2014/6/24 |
| ↓ | 2014/6/27 | ■ex\_exit  　P.40 [デバッガ関係者向け注意事項]に注意事項を追加 | 横山 | - | - |
| ↓ | 2014/7/1 | ■ex\_rh\_evncond  P.150 [デバッガ開発者向け注意事項]  P.151 [マルチコア時の注意点] | ビナイ | 石井  2014/7/3 | 佐々木  2014/7/3 |
| 02 | 2014/7/9 | ■ex\_memread、ex\_memwrite  P.116、P.123 [Return Values]に以下のエラーを追加  ・EX\_SE\_SFLCLKDSB\_ACCFAILED  ・EX\_SE\_SDRAMCLKDSB\_ACCFAILED | 永井 | 石井  2014/7/10 | 佐々木  2014/7/10 |
| 03 | 2014/7/15 | ■ex\_memfill、ex\_memcopy  英訳時の指摘反映  P.130、P.133 [Return Values]にも以下のエラーを追加  ・EX\_SE\_AXICLKDSB\_ACCFAILED  ・EX\_SE\_SFLCLKDSB\_ACCFAILED  ・EX\_SE\_SDRAMCLKDSB\_ACCFAILED | 永井 | - | - |
| ↓ | 2014/7/23 | ■ex\_memread、ex\_memwrite  P.118、P.125  <EX\_M\_DMM\_RD>と<EX\_M\_DMM\_WR>の説明を修正。 | 永井 | - | - |
| ↓ | 2014/7/23 | ■2.5 ターゲット電源に関する動作仕様  P.30  電源OFF、電源OFFからの復帰動作の説明全体を見直し | 永井 | - | - |
| ↓ | 2014/7/23 | ■2.7 非同期デバッグに関する動作仕様  P.34  非同期デバッグは本開発Phaseではサポートしない旨の記載を追加 | 永井 | - | - |
| ↓ | 2014/7/23 | ■表1-1.サポート機能一覧  P.8、P.9  No37　イベント機能-DMAバス検出　Phase2.3開発でのサポートに変更  No58　トレース機能-DMA　Phase2.3開発でのサポートに変更  ■EX\_UNIT\_SLAVEIDSEL  P85「注:本開発Phaseではサポートしない」を追加  P111「注：本開発Phaseでは未サポートのためEX\_UNIT\_SLAVEIDSELを呼び出さないこと。」を追加 | 石井 |  |  |
| 04 | 2014/09/01 | ■2.2.1 起動  P.26,27  PE2 Disable対応のPE確認の手順を追加  ■ex\_socunitinfo2  P.94,105,107  unit\_type=EX\_UNIT\_COREPESELにPE2 Disableに関連した動作およびエラーを追加 | 七尾 | - | - |
| ↓ | 2014/10/02 | ■ex\_setenv  P.77 [Description]の誤記を修正。  ■ex\_socunitinfo2  P.109 unit\_type=EX\_UNIT\_TRCSEL [Argument description]の誤記を修正。 | 横山 | - | - |
| ↓ | 2014/10/08 | ■ex\_socunitinfo2  P.106 [デバッガ関係者向け注意事項] - (2)の表を修正  ■ex\_rh\_evncond  P.146 [Arguments] - acc\_srcを修正  ■ex\_rh\_trceventsel2  P.203 [動作説明] - トレース・データの出力タイプの表を修正 | 横山 | - | - |
| ↓ | 2014/10/17 | ■ex\_trcevent  P.191 [Return Values]を修正  [Description]-[Argument description]を修正  P.192 [動作説明]を修正  [マルチコアおよびスレーブ対応時の注意事項]を修正 | 横山 | - | - |
| ↓ | 2014/10/20 | ■2.1.2 インターフェース関数一覧  P25 表2-1 タイマ・パフォーマンスのI/Fが未サポートのままになっていたのを修正 | 永井 | - | - |
| ↓ | 2014/10/21 | ■全体  ・関数見出しのStatusのCOVの内容を見直し(全体的に"／"に置き換え)  ・取り消し線で見せ消しになっていた不要な記載を削除 | 永井 | - | - |
| ↓ | 2014/10/21 | ■ex\_trcevent  P.192 [デバッガ開発者向け注意事項]を修正 | 横山 | - | - |
| ↓ | 2014/10/27 | ■ex\_rh\_getenv  P.58 [Arguments]を修正  P.59 [Description]-[Argument description]-(2)funcinfo[]を修正  P.60 [デバッガ開発者向け注意事項]を修正  ■ex\_rh\_trctune  P.216 [Arguments]-trctunetypeを修正  P.221 [Arguments]-trc\_tunecondの各種機能タイプ別情報を修正  P.225 [動作説明]を修正 | 横山 | - | - |
| ↓ | 2014/10/29 | ■表紙  P.1 社名をルネサスシステムデザイン株式会社に変更 | 永井 | - | - |
| ↓ | 2014/11/05 | ■3.1 記述規定  P.36 非同期デバッグモード時のStatus説明を追加  ■全体  ・関数見出しのStatusが同期デバッグモード(SYNC)での記載になっていたため、非同期デバッグモード(ASYNC)の記載を追加。  ■ex\_sounitinfo2  P.115 unit\_typeごとのStatus記述を非同期デバッグモードに対応  ■ex\_reset、ex\_setenv、ex\_memwrite  P.46、78、127 非同期デバッグモード時の注意点を記載  ■ 非同期デバッグモードをサポートしない旨の記載削除  P.34、86、114 | 永井 | - | - |
| ↓ | 2014/11/06 | ■ex\_socunitinfo2  P.88 unit\_info[6]に起動時のオプションを追加  P.95 RAM初期化OFFの場合のエラーを追加  P.96 RAM初期化のオプション説明追加  P.97 RAM初期化がオプション化された記載を追加  P.101、103、105 RAM初期化OFFの場合のエラー説明追加  ■ex\_memwrite  P.125 RAM初期化OFFの場合のエラーを追加  P.128 RAM初期化OFFの場合のエラー説明追加  ■ex\_swbrkcond  P.172 RAM初期化OFFの場合のエラーを追加  P.173 RAM初期化OFFの場合のエラー説明追加 | 永井 | - | - |
| ↓ | 2014/11/11 | ■「非同期デバッグに関する動作仕様」  P.34 [非同期デバッグモードにおける制限事項]  ■ex\_socunitinfo2(unit\_type=EX\_UNIT\_ASYNCPEOPT)  P.115 [デバッガ関係者向け注意事項]の記載の一部を[非同期デバッグモード時の注意点]に移動  ■ex\_trcmode  P.189 [マルチコアおよびスレーブ対応時の注意点]の記載の一部を[非同期デバッグモード時の注意点]に移動 | 横山 | - | - |
| ↓ | 2014/11/14 | ■ex\_exit  P.42 [デバッガ関係者向け注意事項]に注意事項を追加 | 永井 | - | - |
| ↓ | 2014/11/14 | ■2.5 ターゲット電源に関する動作仕様  P.30 ＜ターゲット電源OFF＞のステータスに関する記載を追加  ■ex\_getstatus  P.67 [デバッガ開発者向け注意事項]に電源OFF時のトレースステータスの説明を追加 | 永井 | - | - |
| ↓ | 2014/11/14 | ■「非同期デバッグに関する動作仕様」  P.33 「非同期デバッグモードにおけるサポート機能」の表を修正  ■ex\_trcclear  P.190 [Return Values]を修正  [非同期デバッグモード時の注意事項]を追加 | 横山 | - | - |
| ↓ | 2014/11/26 | ■表1-1.サポート機能一覧  P7　Phase2.3向けに変更 | 石井 | - | - |
| ↓ | 2014/11/28 | ■ex\_rh\_getenv  P.62 [Description]-[Argument description]-(2)funcinfo[]の説明を修正 | 横山 | - | - |
| ↓ | 2014/12/1 | ■ex\_rh\_getenv  P.60 [Arguments]-funcinfo[0]を修正  ■ex\_socunitinfo2  P.87 [Arguments]-unit\_typeを修正  P.96 [Arguments]-\*unit\_infoを修正  P.117 [Description]-<unit\_type=EX\_UNIT\_TRACEOPT>を追加 | 横山 | - | - |
| ↓ | 2014/12/2 | ■ex\_rh\_getenv  P.60 [Arguments]-funcinfo[0]を修正、funcinfo[2]を追加  P.62,P.63 [Description]-[Argument description]-funcinf[]を修正  ■ex\_socunitinfo2  P.98 [Arguments]-\*unit\_infoを修正  P.99 [Return Values]を修正  P.117 [Description]-<unit\_type=EX\_UNIT\_SLAVEIDSEL>を修正  P.120 [Description]-<unit\_type=EX\_UNIT\_TRACEMSGPEID>を追加  ■ex\_rh\_evncond  P.155 [Description]-acc\_sizeを修正  ■ex\_rh\_trceventsel2  P.207 [Arguments]-\*item-bit[15:12]を修正  P.208 [Arguments]-\*item-bit23を追加  P.210 [Description]-[動作説明]を修正  ■ex\_rh\_trcbufcont  P.212,213 [Arguments]を修正  P.213 [Description] – [Argument description]を修正 | 横山 | - | - |
| ↓ | 2014/12/11 | ■ex\_rh\_getenv  P.60 [Arguments]を修正  P.63 [Description]-[Argument description]-(2)の「funcinfoの各情報に対して関連するEXEC I/F」の表を修正  ■ex\_rh\_evncond  P.155 [Description]-[Argument description]を修正 | 横山 | - | - |
| ↓ | 2014/12/19 | ■ex\_socunitinfo2  P.120 [デバッガ関係者向け注意事項]-<unit\_type=EX\_UNIT\_TRACEMSGPEID>を修正 | 横山 | - | - |
| ↓ | 2014/12/19 | ■ex\_socunitinfo2  P.98 [Arguments]-\*unit\_infoを修正  ■ex\_rh\_trctune  P.227 [Arguments]を修正 | 横山 | - | - |
| ↓ | 2015/1/14 | ■ex\_rh\_getenv  P.60,61 [Arguments]を修正  P.62,63 [Description]-[Argument description]を修正  ■ex\_rh\_trceventsel2  P.207 [Arguments]を修正 | 横山 | - | - |
| ↓ | 2015/1/19 | ■ex\_memwrite  P.134 [Description]にFlash書き込み中にエラーが発生した場合リセットが発生する旨の記述を追加 | 永井 | - | - |
| ↓ | 2015/2/6 | ■ex\_socunitinof2  ・[Arguments]- EX\_UNIT\_DFLMAPINFOおよびEX\_UNIT\_DFLAREABLOCの追加  ・[Description]- <unit\_type=EX\_UNIT\_DFLMAPINFO>および<unit\_type= EX\_UNIT\_DFLAREABLOC>の追加  ・[Description]- <unit\_type=EX\_UNIT\_DFLBLNKINFO>のBankB対応  ・「unit\_type別EXECの状態と関数の呼出し可(○)/不可(×)の関係」に新規unit\_typeを追加 | 七尾 | - | - |
| ↓ | 2015/2/12 | ■1.4 インターフェースを提供するプログラムについて  ・表1-7、表1-8 サポートOS変更  ・表1-10 ファイルのバージョン表記の更新 | 永井 | - | - |
| ↓ | 2015/2/23 | ・ex\_memcopy  [非同期デバッグモード時の注意点]の誤記修正  ・ex\_regread, ex\_regwrite  「V850 E3v5アーキテクチャ仕様」を用いて文書名を表している記載を関連文書の表記に修正  ・表3-5-2  表中に該当する注釈を記載  ・ex\_trcevent  [デバッガ開発者向け注意事項]の(4)記載の条件を説明文中の文字列に修正 | 七尾 | 七尾  2015/2/24 | 佐々木  2015/2/24 |
| 05 | 2015/4/17 | ・以下のI/F関数の使用していないエラーコードを削除  ex\_initexec、ex\_pincont、ex\_socunitinfo2、  ex\_memcopy、ex\_stop  ・ex\_reset  [Return Values]に記載されていなかったエラーコードがあったため追記  ・ex\_setenv  [Return Values]の使用していないエラーコードを削除。それに伴い、[description]のエラーコードの説明を削除。 | 永井 |  |  |
| ↓ | 2015/5/12 | ・全体  「E2」の記載を「V850E2」に修正  ・2.4 低消費電力モード時の動作仕様  STOPモードが未サポートのままになっていたのでサポートしている旨の記載に修正  ・ex\_getstatus、ex\_setenv、ex\_pinccont、ex\_socunitinfo2、ex\_trcrawmeminfo、ex\_rh\_trctune  PFC1A等のデバイス依存情報の記載を削除。デバイス依存情報はF/W機能仕様書に記載しているため。 | 永井 |  |  |
| ↓ | 2015/5/28 | ・ex\_socunitinfo2  　[Arguments] - unit\_typeを修正  　[Description] - ＜unit\_type=EX\_UNIT\_SLAVEIDSEL＞ –  　[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_trcbufcont  　[マルチコアおよびスレーブ対応時の注意事項]を修正 | 横山 |  |  |
| ↓ | 2015/6/4 | ・ex\_socunitinfo2  [Return Values]にエラーコードを追加  [Description]  ＜unit\_type=EX\_UNIT\_LPDOPT＞ - [デバッガ関係者向け注意事項]を修正  ＜unit\_type=EX\_UNIT\_CG＞ - [Argument description]を修正  ・ex\_rh\_performanceset  [Description] - [Argument description] を修正 | 横山 |  |  |
| ↓ | 2015/6/10 | ・ex\_memwrite  　[フラッシュメモリのBank跨ぎの注意点]を追加 | 永井 |  |  |
| ↓ | 2015/6/15 | ・ex\_rh\_getenv  [Arguments]を修正  ・ex\_socunitinfo2  [Arguments] - <EX\_UNIT\_TRACEMSGPEID>を修正  ・ex\_rh\_trceventsel2  [Arguments]を修正  ・ex\_rh\_performanceset  [デバッガ開発者向け注意事項] を修正 | 横山 |  |  |
| ↓ | 2015/6/16 | ・ex\_rh\_getenv  　[Arguments] - funcinfo[4]を追加  　[Description] - "funcinfoの各情報に対して関連するEXEC I/F"の表にfuninfo[4]を追加  　[デバッガ開発者向け注意事項] - (5)を追加 | 永井 |  |  |
| ↓ | 2015/6/17 | ・1.1 位置付け を修正  ・3.1.1 名称で示される入力範囲規定 を修正  ・ex\_rh\_getenv  [Arguments]を修正  [Descrpition] - [Argument description]を修正  ・ex\_socunitinfo2  [Arguments] - <unit\_type=EX\_UNIT\_SLAVEIDSEL>を修正  [Description] - <unit\_type=EX\_UNIT\_COREPESEL> - [デバッガ関係者向け注意事項]を修正  ・3.2.7 アナライザ – (1)トレース関数  [スレーブトレースの注意事項]を修正  ・ex\_rh\_evncond  [Format]を修正  [Description]  [デバッガ開発者向け注意事項]を修正  [マルチコア時の注意事項]を修正  ・ex\_rh\_trceventsel2  [Arguments]を修正  ・ex\_rh\_performanceset  [Descrition] - [Argument description] - (4)の表を修正 | 横山 |  |  |
| ↓ | 2015/7/17 | ・1.1 位置付け を修正  ・3.1.1 名称で示される入力範囲規定 を修正  ・ex\_rh\_getenv  [Arguments]を修正  [Description] – [Argument description]を修正  ・ex\_socunitinfo2  [Arguments] - <unit\_type=EX\_UNIT\_SLAVEIDSEL>を修正  [Description] - <unit\_type=EX\_UNIT\_COREPESEL> - [デバッガ関係者向け注意事項]を修正  ・ex\_rh\_evncond  [Arguments]を修正  [デバッガ開発者向け注意事項]を修正  [マルチコア時の注意事項]を修正  ・ex\_rh\_evnintgcond  [Description] – [Argument description]を修正  ・3.2.7 アナライザ – (1)トレース関数  [スレーブトレースの注意事項]を修正  ・ex\_rh\_trceventsel2  [Arguments]を修正  ・ex\_rh\_performance  [Description] – [Argument description] - (4)の表を修正 | 横山 |  |  |
| ↓ | 2015/7/23 | 表1-1.サポート機能一覧  　Phase2.4の機能一覧に更新 | 永井 | 永井  2015/8/21 | 佐々木  2015/8/21 |
| 06 | 2015/9/17 | 以下の内容はセキュリティ内容ではなくなったため「RH850G3コア セキュリティMCU用EXECインターフェース変更仕様書」より転記  3.2.2 ハードウェア条件設定  - ex\_getstatus  - ex\_socunitinfo2  + EX\_UNIT\_RSU  + EX\_UNIT\_FID  3.2.3 メモリ操作  - ex\_memwrite  3.2.6 実行  - ex\_resetrun  4.2 インターフェース仕様 | 永井 |  |  |
| ↓ | 2015/10/8 | ・1.1 位置付け を修正  ・3.1.1 名称で示される入力範囲規定 を修正  ・ex\_rh\_getenv  [Arguments]を修正  [Descrpition] - [Argument description]を修正  ・ex\_socunitinfo2  [Arguments] - <unit\_type=EX\_UNIT\_SLAVEIDSEL>を修正  [Description] - <unit\_type=EX\_UNIT\_COREPESEL> - [  デバッガ関係者向け注意事項]を修正  ・ex\_rh\_evncond  [Arguments]を修正  [Description]  [デバッガ開発者向け注意事項]を修正  [マルチコア時の注意事項]を修正  ・ex\_rh\_evnintgcond  [Description] – [Argument description]を修正  ・3.2.7 アナライザ – (1)トレース関数  [スレーブトレースの注意事項]を修正  ・ex\_rh\_trceventsel2  [Arguments]を修正  ・ex\_rh\_performanceset  [Descrition] - [Argument description] - (4)の表を修正 | 横山 |  |  |
| ↓ | 2015/10/13 | ・ex\_swbrkcond  [Format]を修正  [Arguments]を修正  [Description] - [動作説明]を修正  [マルチコア時の注意事項]を修正  [デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2015/10/23 | ・2.7 非同期デバッグに関する動作仕様 を修正  ・ex\_socunitinfo <unit\_type=EX\_UNIT\_FLPGOPT>  [動作説明]を修正 | 横山 |  |  |
| ↓ | 2015/10/30 | ・「2.5 ターゲット電源に関する動作仕様」内の文章修正  　D1x IE850 POD使用時のex\_pincontによるマスク設定の文章を削除(マスク設定に依存しないため削除)  　PODの種類およびコア電源の種類によって、入力されるリセットが異なることの文章を追加 | 有鹿 |  |  |
| ↓ | 2015/11/04 | 文法エラーを修正。 | 有鹿 |  |  |
| ↓ | 2015/11/04 | ・ex\_swbrkcond  [Description] - [動作説明]を修正 | 横山 |  |  |
| ↓ | 2015/11/25 | ・ex\_resetrun  [Description]に[非同期デバッグモード時の注意点]を追加 | 横山 |  |  |
| ↓ | 2015/12/09 | 表1-1.サポート機能一覧  　Phase2.5の機能一覧に更新 | 有鹿  2015/12/09 | 畑原  2015/12/10 | 佐々木  2015/12/10 |
| 07 | 2015/10/21 | ・全体  E2エミュレータの記述を追加  ・1.4.3 動作環境についての注意点  接続可能台数についてE2エミュレータの説明を追加  ・1.4.6 提供ファイルについて  E2エミュレータ対応で追加になるファイルを記載  ・ex\_initexec  E2エミュレータも電源供給機能使用可能である説明を追加  ・ex\_setmeulator  drvtypeにE2/E2 Liteを追加  E2エミュレータはニックネーム機能をサポートする説明を追加  ・ex\_boardinfo2  エミュレータ種別にE2エミュレータを追加  ・6. 強制VDD供給機能の制御仕様  E2エミュレータも電源供給機能使用可能である説明を追加 | 永井 |  |  |
| ↓ | 2015/10/30 | ・1.4.2 ソフトウェアの構成および動作環境について(E2)  　章を追加 | 永井 |  |  |
| ↓ | 2015/11/5 | ・1.4.4 動作環境についての注意点  E2はE2 Liteと合計で4台まで接続可能である注意書きを追加  ・ex\_socunitinfo2(EX\_UNIT\_LPDOPT)  E2エミュレータで設定可能なLPDクロックを追加  ・6章を削除(未使用機能であるため)  ・ex\_exit、ex\_initexec  [デバッガ関係者向け注意事項]の強制VDD供給機能に関する注意事項を削除 | 永井 |  |  |
| ↓ | 2015/12/15 | 表1-1.サポート機能一覧  　Phase2.6の機能一覧に更新 | 永井 | 井出 | 佐々木 |
| ↓ | 2015/12/28 | ・表紙  　関連文書追加  ・1.4.1 ソフトウェアの構成及び動作環境について(E1/E20)  　構成にCommuni.dllを追加  ・2.1.1 インターフェース関数の分類  　「コールバック関数」を「callback\_sts」に修正  ・ex\_exit  　「予定」を削除  ・ex\_rh\_getenv  　[デバッガ開発者向け注意事項]の(5)を修正  ・ex\_getstatus  　[デバッガ開発者向け注意事項]の(4)を削除  ・ex\_stop  　「ユーザに対しては...」の記載を削除 | 永井 | 井出 | 佐々木 |
| 08 | 2016/1/27 | ・ex\_socunitinfo2  [Return Values]にエラーコードを追加  [Description]-<unit\_type=EX\_UNIT\_RSU>-[動作説明]を修正 | 横山 |  |  |
| 09 | 2016/04/26 | F1K(2ndCut)対応の改定に伴う変更  ・1．概要 表1－1  　V3.0欄を追加、サポートマイコンのF1Kの  　備考に1stCut未サポートの説明を追加 | 有鹿 |  |  |
| ↓ | 2016/04/26 | 強制リセット機能の対応に伴う変更  ・ex\_reset  　mode(EX\_F\_RESET)を新規追加  　mode(EX\_F\_RESET)の説明、注意事項を追加 | 有鹿 |  |  |
| ↓ | 2016/05/12 | ・ex\_rh\_getenv  [Arguments]を修正  [Argument description]を修正  ・ex\_pincont  [Arguments]を修正  [動作説明]を修正  [デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2016/05/13 | ・1.概要を修正  ・ex\_reset  [Argument description]を修正  ・ex\_rh\_getenv  [Arguments]を修正  [デバッガ開発者向け注意事項]を修正  ・ex\_getstatus  [Arguments]を修正  [Return Values]を修正  [Argument description]を修正  ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_FLMKOP>  [動作説明]を修正  ・ex\_rh\_evncond  [Description]を修正  [デバッガ開発者向け注意事項]を修正  ・ex\_trcmode  [デバッガ関係者向け注意事項]を修正  ・4.2 インターフェース仕様  コールバック関数  [Argument description]を修正  [デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2016/05/17 | ・ex\_pincont  [Arguments]を修正  [動作説明]を修正 | 横山 |  |  |
| ↓ | 2016/05/24 | ・ex\_exit  　[動作説明]に終了時の動作説明を追加 | 永井 |  |  |
| ↓ | 2016/05/30 | ・サポート機能一覧にワークRAM領域の設定/参照を追加  ・ex\_socunitinfo2  　unit\_typeにEX\_UNIT\_WORKRAMを追加  エラーコードEX\_PE\_WORKRAMRANGEを追加  　EX\_UNIT\_WORKRAMの動作説明と注意事項を記載 | 新井 |  |  |
| ↓ | 2016/05/30 | ・ex\_memwrite  　modeにEX\_M\_FLASH\_P\_WRを追加  　EX\_M\_FLASH\_P\_WRの動作説明と注意事項を記載 | 新井 |  |  |
| ↓ | 2016/06/22 | ・2.1.1起動と終了  PiggyBoard対応の追加。  ex\_socunitinfo2<unit\_type=EX\_UNIT\_LPDOPT>に  端子リセットディレイ対策の必要性を追加  ・3.2.2 ハードウェア条件設定  PiggyBoard対応の追加。  ex\_socunitinfo2 <unit\_type=EX\_UNIT\_LPDOPT>  unit\_info[7]にコールドスタートシーケンスの手順指定  を追加 | 七尾 |  |  |
| ↓ | 2016/6/28 | ・3.2.2 ex\_socunitinfo2  　unit\_type EX\_UNIT\_WORKRAMの仕様を変更  　　CPU1のローカルRAMまたはプライマリローカルRAMがない  デバイスの場合、リテンションRAMを用いる  上記に伴い開始アドレス初期値の記述を変更  ローカルRAMSelf領域指定時の動作を追加 | 新井 |  |  |
| ↓ | 2016/7/20 | 文書番号をIDF3体系からRSD-DMS体系に変更 | 七尾 |  |  |
| ↓ | 2016/7/26 | 3.2.2　ex\_getstatus hwbrfact各ブレーク要因の発生状態  以下の説明を追記。  同時に発生した場合は優先順位の高い要因を格納する。  優先順位については、表 4-1参照。  3.2.6　ex\_hwbrkevent  [デバッガ開発者向け注意事項]欄にG3MHコア不具合の説明追記。 | 畑原 |  |  |
| ↓ | 2016/07/27 | 表紙  　IDF番号取得に伴いIDF番号修正 | 有鹿 | 畑原 | 佐々木 |
| ↓ | 2016/08/09 | 1.改定履歴  2016/07/26版の誤記を訂正。  　誤)表5-1 正)表4-1  2.Phase3.0をV3.0に訂正。 | 有鹿 | 畑原 | 佐々木 |
| 10 | 2016/5/24 | ・新規I/Fとしてex\_getemulistを追加  追加箇所は以下  1.1 位置付け  2.1.1 インターフェース関数の分類  2.1.2 インターフェース関数一覧  2.2.1 起動  3.2.1 起動、終了、リセット | 永井 |  |  |
| ↓ | 2016/6/7 | 1.1 位置付け のEXEC仕様関連表を修正  2.1.1 インターフェース関数の分類 を修正  2.1.2 インターフェース関数一覧 を修正  2.7 非同期デバッグに関する動作仕様 を修正  2.8 外部ソフトトレース単線出力機能に関する動作仕様 を追加  ・ex\_reset [Return Values]、[動作説明]を修正  ・ex\_rh\_getenv [Arguments]、[Return Values]、[Argument description]、[動作説明]を修正  ・ex\_getstatus [Return Values]、[動作説明]を修正  ・ex\_boardinfo2 [Return Values]、[動作説明]を修正  ・ex\_pincont [Retrun Values]、[動作説明]を修正  ・ex\_socunitinfo2 [Arguments]、[Return Values]を修正  　<unit\_type=EX\_UNIT\_RUNCONTOPT> [動作説明]を修正  　<unit\_type=EX\_UNIT\_TRCSEL> [Argument description]、[動作説明]、[デバッガ関係者向け注意事項]を修正  　<unit\_type=EX\_UNIT\_VIRTUALINFO> [動作説明]を修正  ・ex\_memread [Return Values]、[動作説明]を修正  ・ex\_memwrite [Return Values]、[動作説明]を修正  ・ex\_memfill [Return Values]、[動作説明]を修正  ・ex\_regread [Return Values]、[Description]を修正  ・ex\_regwrite [Return Values]、[Description]を修正  ・ex\_rh\_evncond [Return Values]、[動作説明]を修正  ・ex\_rh\_evnintgcond [Return Values]、[動作説明]を修正  ・ex\_step [Return Values]、[Description]を修正  ・ex\_hwbrkevent [Return Values]、[Description]を修正  ・ex\_swbrkcond [Return Values]、[動作説明]を修正  ・ex\_trcrun [Return Values]、[動作説明]を修正  ・ex\_trcmode [Return Values]、[動作説明]を修正  ・ex\_trcclear [Return Values]、[動作説明]を修正  ・ex\_trcevent [Return Values]、[動作説明]を修正  ・ex\_trcdelay [Return Values]、[動作説明]を修正  ・ex\_rh\_trceventsel2 [Return Values]、[動作説明]を修正  ・ex\_rh\_trcbufcont [Return Values]、[動作説明]を修正  ・ex\_trcrawmeminfo [Return Values]、[動作説明]を修正  ・ex\_trcrawmemread [Return Values]、[動作説明]を修正  ・ex\_rh\_trchwevncopeinfo [Return Values]、[動作説明]を修正  3.2.7に(2)外部ソフトトレース単線出力関数を追加  ・ex\_timeevent2 [Return Values]、[動作説明]を修正  ・ex\_rh\_timeevent [Return Values]、[動作説明]を修正  ・ex\_rh\_performanceset [Return Values]、[動作説明]を修正 | 横山 |  |  |
| ↓ | 2016/6/17 | 2.8 [主な制限事項]を修正  ・ex\_rh\_sfttrctune  [Arguments]、  [Description]<filter\_type=EX\_FILTERTYPE\_PC>を修正 | 横山 |  |  |
| ↓ | 2016/6/24 | ・ex\_rh\_getenv  [Arguments]、[Argument description]を修正  ・ex\_rh\_trctune  [Arguments]、[Retrun Values]、  [動作説明]<trc\_tunetype=EX\_TRC\_TRCOUTIFOPTION>を修正、<trc\_tunetype= EX\_TRC\_PODFIFOCOUNT>を追加 | 横山 |  |  |
| ↓ | 2016/7/26 | ・新規I/Fとしてex\_getemulistを追加  追加箇所は以下  1.1 位置付け  2.1.1 インターフェース関数の分類  2.1.2 インターフェース関数一覧  2.2.1 起動  3.2.1 起動、終了、リセット  ・ex\_exitの終了モードにEX\_END\_PROC\_WO\_RESETを追加 | 永井 |  |  |
| ↓ | 2016/8/30 | ・ex\_rh\_trctune  [Return Values]、[動作説明] <trc\_tunetype= EX\_TRC\_PODFIFOCOUNT>を修正 | 横山 |  |  |
| ↓ | 2016/9/15 | ・ex\_pincont  [動作説明]を修正 | 横山 |  |  |
| ↓ | 2016/10/6 | ・ex\_rh\_getenv  [Arguments]を修正  ・ex\_rh\_trctune  [Arguments]、[動作説明] <trc\_tunetype= EX\_TRC\_TRCRAMOPTION >を修正 | 横山 |  |  |
| ↓ | 2016/10/07 | エミュレータ接続検出用デバッグ情報出力機能対応  ・ex\_rh\_getenv  　[Arguments]‐funcinfo[7]を新規追加  　funcinfoの各情報に対して関連するEXEC I/Fの表に  　funcinfo[7]を新規追加  ・ex\_socunitinfo2  　[Arguments]‐unit\_typeにEX\_UNIT\_DBGINFOUTを追加  　[Arguments]‐unit\_infoにEX\_UNIT\_DBGINFOUTを追加  　[Description]にEX\_UNIT\_DBGINFOUTを追加  　[Description]‐unit\_type別 EXECの状態と関数の呼出し可(○)/不可(×)の関係のNo.31を新規追加 | 有鹿 |  |  |
| ↓ | 2016/10/13 | ・ex\_extpower  [Arguments]のmode、valに設定可能な値を追加  [Description]の[Argument description]にmodeパラメータの一覧と動作の関係を示す表を追加 | 永井 |  |  |
| 11 | 2016/10/07 | RH850/G4世代の仕様を追加  2.2.1 起動  　・デバイスの品種によって自走しないで起動処理を行なう記載を追加  　・起動手順にex\_socunitinfo2(EX\_UNIT\_FID)を追加  ex\_exit  　・[デバッガ関係者向け注意事項]に、デバイスの品種にって必ずデバッグ通信I/Fモードが初期状態(JTAG)に戻ることの記載を追加  ex\_socunitinfo2  　・Phaseの記載を削除  　・<unit\_type=EX\_UNIT\_LPDOPT>の[動作説明]に、デバイスの品種によってコマンド発行時の指定を無視することの記載を追加  ・<unit\_type=EX\_UNIT\_RSU>の[動作説明]にライタモードへ遷移させずにLPD接続処理を行なうことの記載を追加 | 有鹿 |  |  |
| ↓ | 2016/10/07 | RH850/G4世代の仕様を追加  ex\_rh\_getenv  ・モジュール搭載情報にG4世代MCUのビット配置を追記  　・トレースメッセージタイプにG4M-Hタイプを追記  　・RCUのモニタ領域にG4世代MCUの場合を追記  ex\_socunitinfo2  　・<EX\_UNIT\_ASYNCPEOPT>にG4世代MCUのビット配置を追記  ex\_memread, ex\_memwrite, ex\_memfill  ・メモリアクセス手段の格納値にG4世代MCUのビット配置を追記  ex\_rh\_evncond  　・G4世代MCUで設定可能なイベント条件、検出器個数の表を追加  ex\_rh\_trceventsel2  　・トレース・データ出力タイプの指定にG4世代MCUのタイプを追加  ex\_rh\_performanceset  　・G4コアのパフォーマンス計測項目の表を追加  3.4 レジスタID  　・ベクトルレジスタにG4MHコアのベクトルレジスタを追加  4.2 インターフェース仕様  　・コールバック関数の＜status\_type=EX\_WM\_BRKSTAT指定時＞にG4世代MCUのコア番号ビット配置を追加 | 島田 |  |  |
| ↓ | 2016/10/18 | RH850/G4世代の仕様を追加  ・ex\_socunitinfo2  [Description]にEX\_UNIT\_DBGINFOUTを追加  デバッガ関係者向け注意事項にG4世代の仕様を追加 | 有鹿 |  |  |
| ↓ | 2016/10/18 | 1.　概要  　サポート機能一覧表にRH850 G4世代の表を追加  ・ex\_rh\_evncond  　[Description]にG4世代ではEX\_RUNB\_RWND\_STS、  EX\_RUNB\_RND\_STS、EX\_RUNB\_WND\_STSをサポートしないことを追記。 | 島田 |  |  |
| ↓ | 2016/10/25 | 表 3 4 G3世代MCUのstat条件とイベント条件の対応  　表内「△1」の注記を「※」注記へ変更 | 島田 |  |  |
| ↓ | 2016/11/07 | ex\_socunitinfo2 <EX\_UNIT\_FID>  ・unit\_info[0]～[7]がG3世代ではCodeFlashセキュリティIDコード、G4世代ではCustomerIDコード入力用に使用されることを明記  ex\_memwrite  　・EX\_M\_FLASH\_PV\_WRの書き換え対象にフラッシュエキストラ領域を追加  I/F全般  　・G4世代デバイスファイルフォーマットの方針によりICUMコアがPE8→PE7へ変更となり、I/F仕様書内の記述を修正。 | 島田 |  |  |
| ↓ | 2016/11/18 | 翻訳依頼 | 畑原 |  |  |
| ↓ | 2016/11/21 | ex\_memwrite  　・EX\_M\_FLASH\_PV\_WRのフラッシュエキストラ領域への書き換えバイト数の単位としてID格納エリアについては32byte単位で書き換えることを追記  ex\_socunitinfo2 <EX\_UNIT\_FID>  ・C-TEST-ID、RHSIF-ID、Serial Programmer-IDの認証解除IDを追加  2.2.1　起動  　・起動処理中にC-TEST-ID、RHSIF-ID、Serial Programmer-IDの  　　認証必要可否を判定する方法を追記 | 島田 |  |  |
| ↓ | 2016/11/22 | 1.1 位置付け のEXEC仕様関連表を修正  2.1.1 インターフェース関数の分類 を修正  2.1.2 インターフェース関数一覧 を修正  2.7 非同期デバッグに関する動作仕様 を修正  2.8 [主な制限事項]を修正  ・ex\_rh\_getenv  [Arguments]を修正  ・ex\_socunitinfo2  [Return Values]、[デバッガ関係者向け注意事項]を修正  ・ex\_step  [Return Values]、[Description]を修正  3.2.7 (2)外部ソフトトレース単線出力関数を修正  ・ex\_rh\_sfttrcsw  [Format]、[Arguments]、[Argument description]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcrunを削除  ・ex\_rh\_sfttrcmode  [Format]、[Arguments] 、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sftrcfilter  [Arguments]を修正  <filter\_type=EX\_FILTERTYPE\_PC> - [デバッガ開発者向け注意事項]を修正  <filter\_type=EX\_FILTERTYPE\_TAG> - [デバッガ開発者向け注意事項]を修正  <filter\_type=EX\_FILTERTYPE\_REG> - [デバッガ開発者向け注意事項]を修正  ・ex\_rh\_sfttrctune  <tune\_type=EX\_TUNETYPE\_RATE> - [デバッガ開発者向け注意事項]を修正  <tune\_type=EX\_TUNETYPE\_OUTPUT> - [デバッガ開発者向け注意事項]を修正  <tune\_type=EX\_TUNETYPE\_FORMAT> - [デバッガ開発者向け注意事項]を修正  5.2 INIファイル仕様 表5-2を修正 | 横山 |  |  |
| ↓ | 2016/12/09 | 2.2.1　起動  　・起動処理中にC-TEST-ID、RHSIF-ID、Serial Programmer-IDの  　　認証必要可否を判定する方法の追記を見せ消しに変更  ex\_socunitinfo2 <EX\_UNIT\_FID>  　・numに指定する値を18以上に修正  　・unit\_info[44:19]について本開発で指定されても  未使用である旨の記載を追加  　・引数説明のC-TEST ID認証、RHSIF ID認証、Serial Programmer  IDについて、本開発では指定されても認証処理は行わず無視  する旨の記載を追加 | 有鹿 |  |  |
| ↓ | 2017/1/10 | ex\_regread,ex\_regwrite  　・128bitレジスタのリード/ライトデータ格納順の説明を修正  　　(内容に変更はなく、詳細な説明を追加) | 島田 |  |  |
| ↓ | 2017/1/10 | 翻訳Grからの指摘を和文へ反映  ex\_rh\_getenv  　・「非CPUマスタDMA」を「非CPUマスタ(DMA)」へ記述変更  ex\_socunitinfo2 <EX\_UNIT\_WORKRAM>  ・「CPU1/CPU0」を「CPU@PE1/PE0」へ記述変更  ex\_rh\_performanceset  　・表3-16の下に『※1』の文章を追加 | 島田 |  |  |
| ↓ | 2017/1/11 | ex\_socunitinfo2  　・EX\_UNIT\_FLLBINFOをG4では未サポートする旨を記載 | 新井 |  |  |
| ↓ | 2017/01/11 | 翻訳Grからの指摘を和文へ反映  表1-2 No.84  　・用語の修正(フェッチ停止⇒初期停止)  全体  　・G4M-Hの用語をG4MHに統一  2.5章　本文  　・I/F名の誤記修正(ex\_getstat⇒ex\_getstatus)  ex\_exit [デバッガ関係者向け注意事項] (2)  　・I/Fパラメータ名の誤記修正(EX\_LPDOPT⇒EX\_UNIT\_LPDOPT)  ex\_rh\_getenv [Arguments] mcuinfo[1]  ・未使用ビット名の誤記修正(b20-b31⇒b19-b31)  ex\_socunitinfo2(unit\_type=EX\_UNITFID) [引数説明] (5)  ・格納する配列番号の誤記修正(42⇒43 , 43⇒44)  ex\_socunitinfo2(unit\_type=EX\_UNIT\_FID)　[デバッガ関係者向け注意事項]  　・[CodeFlashID認証]の章にCustomerIDの用語を追加  　・[Serial Programmer ID認証]の章の配列番号の誤記修正  ex\_socunitinfo2(unit\_type=EX\_UNIT\_DBGINFOUT) 　[デバッガ関係者向け注意事項]  ・[RH850 G4世代]の記載文章を赤字に変更  ex\_memread [デバッガ関係者向け注意事項]  ・F/W機能仕様書名を修正  ex\_memwrite <フラッシュメモリの書き込み単位> (3)フラッシュエキストラ領域  　・メモリダンプイメージの例の番号の誤記修正(⑤⇒④)  ex\_socunitinfo2  　・Phaseの記載を削除  ・unit\_type=EX\_UNIT\_LPDOPT,EX\_UNIT\_RSUの[動作説明]に  反映が漏れていた文言を再度追加 | 有鹿 |  |  |
|  | 2017/02/08 | Word2016で再保存する。 | 横山 |  |  |
|  | 2017/02/20 | ・2.8章　E2エミュレータ拡張機能に関する動作仕様  　新規追加  ・全体  　E2拡張機能有効時のRUN中呼び出しエラーを追加  ・ex\_getstatus  　拡張機能のブレーク要因を追加  　拡張機能有効時の注意点を追加 | 永井 |  |  |
| ↓ | 2017/2/21 | 1.1 EXEC仕様関連表を修正  2.1.1 [外部ソフトトレース単線出力制御関数]を修正  2.1.2 インターフェース関数一覧を修正  2.8 [機能概要]、[主な制限事項]を修正  2.9 表を修正  ・ex\_getstatus  [Arguments]、[Return Value]、[Description]-[引数説明]、[マルチコア時の注意点]、[デバッガ開発者向け注意事項]、[非同期デバッグモード時の注意点]を修正  ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_TRCSEL> [デバッガ関係者向け注意事項]を修正  3.2.7を修正  3.2.7 (2)外部ソフトトレース単線出力関数を修正  ・ex\_rh\_sfttrcrunを追加  ・ex\_rh\_sfttrcrawmemmodeを追加  ・ex\_rh\_sfttrcrawmeminfoを追加  ・ex\_rh\_sfttrcrawmemreadを追加 | 横山 |  |  |
| ↓ | 2017/02/22 | 1.1 位置付け のEXEC仕様関連表を修正  2.1.2 インターフェース関数一覧を修正  2.7 サポート機能一覧、[サポートしない機能について]を修正  2.8と2.9を入れ替え  2.9 項タイトル、[機能概要]、[主な制限事項]、[非同期デバッグモード時の注意点]を修正  ・ex\_reset [Return Values]、[動作説明]を修正  ・ex\_rh\_getenv [Arguments]、[Return Values]、[動作説明]を修正  ・ex\_getstatus [Arguments]、[引数説明]、[マルチコア時の注意点]、[デバッガ開発者向け注意事項]、[非同期デバッグモード時の注意点]を修正  ・ex\_rh\_getenv [Return Values]、[動作説明]を修正  ・ex\_boardinfo2 [Return Values]、[動作説明]を修正  ・ex\_pincont [Return Values]、[動作説明]を修正  ・ex\_socunitinfo2 [Arguments]、[Return Values]  <unit\_type=EX\_UNIT\_TRCSEL> [動作説明]、[デバッガ関係者向け注意事項]を修正、[非同期デバッグモード時の注意点]を追加  < unit\_type=EX\_UNIT\_VIRTUALINFO> [動作説明]を修正  ・ex\_memread [Return Values]、[動作説明]を修正  ・ex\_memwrite [Return Values]、[動作説明]を修正  ・ex\_memfill [Return Values]、[動作説明]を修正  ・ex\_regread [Return Values]、[動作説明]を修正  ・ex\_regwrite [Return Values]、[動作説明]を修正  ・ex\_rh\_evncond [Return Values]、[動作説明]を修正  ・ex\_rh\_evnintgcond [Return Values]、[動作説明]を修正  ・ex\_step [Return Values]、[動作説明]を修正  ・ex\_hwbrkevent [Return Values]、[動作説明]を修正  ・ex\_swbrkcond [Return Values]、[動作説明]を修正  ・ex\_trcrun [Return Values]、[動作説明]を修正  ・ex\_trcmode [Return Values]、[動作説明]を修正  ・ex\_trcclear [Return Values]、[動作説明]を修正  ・ex\_trcevent [Return Values]、[動作説明]を修正  ・ex\_trcdelay [Return Values]、[動作説明]を修正  ・ex\_rh\_trceventsel2 [Return Values]、[動作説明]を修正  ・ex\_rh\_trcbufcont [Return Values]、[動作説明]を修正  ・ex\_trcrawmeminfo [Return Values]、[動作説明]を修正  ・ex\_trcrawmemread [Return Values]、[動作説明]を修正  ・ex\_rh\_trchwevncopeinfo [Return Values]、[動作説明]を修正  ・ex\_rh\_sfttrcsw [Status]、[Function]、[Format]、[Arguments]、[Return Values]、[動作説明]、[デバッガ関係者向け注意事項]、[非同期デバッグモード時の注意点]を修正  ・ex\_rh\_sfttrcmode [Status]、[Return Values]、[動作説明]、[デバッガ関係者向け注意事項] 、[非同期デバッグモード時の注意点]を修正  ・ex\_rh\_sfttrcfilter [Status]、[Return Values]、[動作説明]、[デバッガ関係者向け注意事項]  <filter\_type=EX\_FILTERTYPE\_PC> [非同期デバッグモード時の注意点]を修正  <filter\_type=EX\_FILTERTYPE\_TAG> [動作説明]、[デバッガ関係者向け注意事項]、[非同期デバッグモード時の注意点]を修正  <filter\_type=EX\_FILTERTYPE\_REG> [動作説明]、[デバッガ関係者向け注意事項]、[非同期デバッグモード時の注意点]を修正  ・ex\_rh\_sfttrctune [Status]、[Return Values]を修正  <tune\_type=EX\_TUNETYPE\_RATE> [動作説明]、[デバッガ関係者向け注意事項]を修正  <tune\_type=EX\_TUNETYPE\_OUTPUT> [動作説明]、[デバッガ関係者向け注意事項]を修正  <tune\_type=EX\_TUNETYPE\_FORMAT> [動作説明]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcrun [Functio Name]、[Status]、[Return Values]、[動作説明]を修正、[非同期デバッグモード時の注意点]を追加  ・ex\_rh\_sfttrcrawmemmode [Status]、[Return Values]、[動作説明]、[デバッガ関係者向け注意事項]を修正、[非同期デバッグモード時の注意点]を追加  ・ex\_rh\_sfttrcrawmeminfo [Status]、[Fromat]、[Return Values]、[動作説明]、[デバッガ関係者向け注意事項]を修正、[非同期デバッグモード時の注意点]を追加  ・ex\_rh\_sfttrcrawmemread [Status]、[Return Values]、[動作説明]を修正、[非同期デバッグモード時の注意点]を追加  ・ex\_timeevent2 [Return Values]、[動作説明]を修正  ・ex\_rh\_timeevent [Return Values]、[動作説明]を修正  ・ex\_rh\_performanceset [Return Values]、[動作説明]を修正 | 横山 |  |  |
| ↓ | 2017/02/23 | ・全体  　エラーコード「EX\_SE\_CANTCALL\_ASPRUNNING」の説明を修正  ・ex\_getstatus [拡張機能有効時の注意点]を修正  ・以下のI/F関数でエラー「EX\_SE\_CANTCALL\_ASPRUNNING」を返す仕様を追加  　- ex\_rh\_sfttrcsw  　- ex\_rh\_sfttrcmode  　- ex\_rh\_sfttrcevent  　- ex\_rh\_sfttrctune | 永井 |  |  |
| ↓ | 2017/03/02 | ・表4-1　拡張機能のブレーク要因を追加 | 永井 |  |  |
| ↓ | 2017/03/06 | 2.9 項タイトル、[機能概要]、[サポート機能]を修正  ・ex\_rh\_getenv [Arguments]、[Description]-[引数説明]を修正  ・ex\_rh\_sfttrcmode [Arguments]、[マルチコア対応時の注意点]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcfilter EX\_FILTERTYPE\_PC [マルチコア対応時の注意点]を修正  ・ex\_rh\_sfttrcfilter EX\_FILTERTYPE\_TAG [マルチコア対応時の注意点]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcfilter EX\_FILTERTYPE\_REG [マルチコア対応時の注意点]を修正  ・ex\_rh\_sfttrctune EX\_TUNETYPE\_RATE [マルチコア対応時の注意点]を修正  ・ex\_rh\_sfttrctune EX\_TUNETYPE\_OUTPUT [マルチコア対応時の注意点]を修正  ・ex\_rh\_sfttrctune EX\_TUNETYPE\_FORMAT [マルチコア対応時の注意点]を修正 | 横山 |  |  |
| ↓ | 2017/03/07 | RH850 EXEC V4.00.00開発 機能設計PDR完了。 | 有鹿 | 畑原 | 佐々木 |
| ↓ | 2017/03/21 | ・ex\_socunitinfo2  　エラーコードからEX\_SE\_SFTTRCRUNを削除  ・翻訳依頼 | 横山 | 畑原 | 佐々木 |
| ↓ | 2017/03/29 | ・INIファイル仕様 LogFileNameの書式追加 | 緒方 |  |  |
| ↓ | 2017/04/11 | ・ex\_swbrkcond [Return Values]、[Description]-[動作説明]を修正  ・ex\_rh\_trctune [Return Values]、[Description]-[動作説明]を修正  ・ex\_timeread2 [Return Values]、[Description]-[動作説明]を修正 | 横山 |  |  |
| ↓ | 2017/04/27 | 外部ソフトトレース単線出力機能のデバイス不具合に対応するため  以下の改定を実施。  1.ex\_rh\_sfttrcsw  ・エラー(EX\_SE\_ALREADY)を削除  ・エラー(EX\_DE\_ALREADYSFTCORESET)を新規追加  2.ex\_rh\_sfttrcrawmeminfo、ex\_rh\_sfttrcrawmemread  エラー(EX\_DE\_SFTCANTREC)を新規追加 | 畑原 |  |  |
| ↓ | 2017/05/11 | 2.9 [サポート機能]を修正  ・ex\_rh\_getenv [Arguments]を修正  ・ex\_socunitinfo2 [Arguments]、<EX\_UNIT\_TRCSEL>-[引数説明]を修正  ・ex\_trcmode [Format]を修正  ・ex\_trcevent [Format]を修正  ・ex\_trcdelay [Format]を修正  ・ex\_rh\_trceventsel2 [Format]を修正  ・ex\_rh\_trcbufcont [Format]を修正  ・ex\_rh\_trctune [Format]を修正  3.2.7 (2)を修正  ・ex\_rh\_sfttrcmode [Description]-[引数説明]、[マルチコア対応時の注意点]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcfilter [Arguments]、[引数説明]、[動作説明]、[マルチコア対応時の注意点]、[デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrctune [マルチコア対応時の注意点]を修正  ・ex\_rh\_sfttrcrun [動作説明]を修正  ・ex\_rh\_sfttrcrawmemmode [Format]、[Description]-[引数説明]を修正  ・ex\_rh\_sfttrcrawmeminfo [デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcrawmemread [Format]を修正  ・ex\_covmapping [Format]を修正  ・ex\_covmapping2 [Format]を修正 | 横山 |  |  |
| ↓ | 2017/5/12 | ・図1.2 RH850G3 からRH850G3/G4に訂正  ・表3-3と表3-4の表番号を訂正  ・2.7 非同期デバッグに関する動作仕様  [サポートしない機能について]  外部ソフトトレース単線出力機能はサポートしない。  との記載に訂正。 | 畑原 |  |  |
| 11 | 2017/5/12 | ・2.9 [主な制限事項]欄(3),(4)に外部ソフトトレース単線出力  機能の制限事項追加。  ・RH850 EXEC V4.00.00開発 UTB工程での機能設計クローズ | 有鹿 | 畑原 | 佐々木 |
| 12 | 2017/6/21 | V4.00.00での修正履歴を削除 | 横山 |  |  |
| ↓ | 2017/6/21 | RTOSタスクアナライザ対応  ・ex\_rh\_trceventsel2 [Arguments]を修正 | 横山 |  |  |
| ↓ | 2017/6/22 | G4世代外部ソフトトレース単線出力対応  ・ex\_rh\_getenv [Arguments]を修正  ・ex\_rh\_sfttrcsw [Arguments]を修正  ・ex\_rh\_sfttrcrawmemread [デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2017/7/06 | G4世代MCUのロックステップエラー抑止対応  ・2.2.1の本文に説明文追加  ・ex\_reset [動作説明]に説明文追加 | 有鹿 |  |  |
| ↓ | 2017/7/7 | 誤記訂正・用語統一  ・ex\_rh\_sfttrcrawmemmode [Description]-[引数説明]を修正  ・ex\_rh\_sfttrcrawmeminfo [デバッガ関係者向け注意事項]を修正  ・ex\_rh\_sfttrcrawmemread [Description]-[引数説明]、[デバッガ関係者向け注意事項]を修正  ・ex\_timeread2 [Description]-[動作説明]を修正  履歴の誤記を修正 | 横山 |  |  |
| ↓ | 2017/7/6 | G4世代MCU 認証用IDコード対応  ・2.2.1章  　G3世代MCUの起動手順とG4世代MCUの起動手順を分割して記載するように変更  ・ex\_socunitinfo2章の以下の部分を追加/変更  　[Arguments]  unit\_type欄  EX\_UNIT\_FIDの説明の記載追加  EX\_UNIT\_CMFID,EX\_UNIT\_OPTIDの説明を新規追加  　unit\_info欄  EX\_UNIT\_FIDの説明の記載修正(G4世代MCUに関する記載削除)  EX\_UNIT\_CMFID,EX\_UNIT\_OPTIDの説明を新規追加 | 有鹿 |  |  |
| ↓ | 2017/7/11 | G4世代MCU 認証用IDコード対応  ・2.2.1章  　起動手順完了後の補足説明を追加 | 有鹿 |  |  |
| ↓ | 2017/7/18 | G4世代MCU 認証用IDコード対応  ・2.2.1章、ex\_socunitinfo2章のEX\_UNIT\_CMFID,EX\_UNIT\_OPTIDについての説明をDFファイルのindexに関する説明を修正 | 有鹿 |  |  |
| ↓ | 2017/7/21 | 1. 表1-1を修正。 | 横山 |  |  |
| ↓ | 2017/08/03 | G4世代MCU TEU\_CPUの命令系範囲イベントデバイス不具合対応  ・ex\_rh\_getenv　[Arguments]-funcinfo[2]の説明追加  [Description]-[引数説明]-[funcinfoの各情報に対して関連するEXEC I/F]-funcinfo[2]の説明追加  ・ex\_rh\_evncond　表3-8に注記(6,7)を追加 | 有鹿 |  |  |
| ↓ | 2017/08/03 | G4世代MCU CPUコアデータアクセスイベントのエラー応答対応  ・ex\_rh\_evncond  [Description]-[引数説明]-statにエラー応答する旨の記載を追加  表3-8　注記(※2)と(※3)を合わせた上で記載内容を修正(以降の注記番号も修正) | 有鹿 |  |  |
| ↓ | 2017/7/31 | G4世代スレーブトレース対応  1.1 表1-5を修正  1.2.1 表を修正  ・ex\_rh\_getenv [Arguments]、[デバッガ開発者向け注意事項]を修正  ・ex\_socunitinfo2 [Arguments]を修正  　　EX\_UNIT\_COREPESEL [デバッガ向け注意事項]の表を修正  　　EX\_UNIT\_TRCSEL [デバッガ向け注意事項]を修正  ・ex\_rh\_evncond [デバッガ向け注意事項]、[マルチコア時の注意事項]を修正  3.2.7(1) [スレーブトレースの注意事項]、[各I/Fごとの設定対象]を修正  ・ex\_rh\_trceventsel2 [Arguments]、[動作説明]を修正  ・ex\_rh\_trcbufcont [Arguments]、[Description]-[引数説明]を修正 | 横山 |  |  |
| ↓ | 2017/08/07 | 誤記訂正  ・ex\_getstatus [動作説明]、[デバッガ開発者向け注意事項]を修正  4.2 [Description]-[引数説明]を修正 | 横山 |  |  |
| ↓ | 2017/08/08 | ・ex\_initexecにエラーコード「EX\_UE\_NOSPRT\_IE850」を追加し、[動作説明]にエラーの説明を追加。  ・ex\_socunitinfo2にエラーコード「EX\_PE\_NOSPRT\_LPD」を追加し、EX\_UNIT\_LPDOPTの動作説明にエラーの説明を追加。  ・ex\_socunitinfo2のunit\_type別呼び出し可否の表を変更 EX\_UNIT\_RSUとEX\_UNIT\_FIDをRUN中呼び出し可に変更  ・ex\_resetrunを同期デバッグモードでも呼び出し可に変更 | 永井 |  |  |
| ↓ | 2017/08/09 | G4世代E2M(製品チップ)対応  ・1．表1－2　V4.01欄、備考欄の記載追記 | 有鹿 |  |  |
| ↓ | 2017/08/25 | ・RH850G4 イベント機能対応  ex\_rh\_evncondのacc\_srcにRH850G4のTEUユニット毎のアクセスマスタ選択を追加  ・IE850A対応  表紙 対応エミュレータにIE850Aを追加  1.概要 IE850 Next(仮称)をIE850Aに修正  1.2.1 記述説明にIE850Aを追加  1.4.2 ソフトウェアの構成および動作環境について(E2)にIE850Aの記述を追加  1.4.4 動作環境についての注意点にIE850Aを追加  1.4.7 提供ファイルについてにIE850AのBFW/FPGAファイルを追加  2.2.1 起動の電源供給設定に関してIE850Aを追加  2.3 CPUリセットが発生する関数の対象にIE850Aを追加  2.9 外部ソフトトレース単線出力機能に関する動作仕様にIE850Aを追加  ex\_exit EX\_END\_PROC\_WO\_RESETサポートにIE850Aを追加  ex\_initexec 電源供給サポートエミュレータにIE850Aを追加  ex\_setemulatorにIE850Aを追加  ex\_getemulistにIE850Aを追加  ex\_extpowerにIE850Aの電源供給仕様を追加  ex\_boardinfo2にIE850Aを追加  ex\_socunitinfo2 EX\_UNIT\_LPDOPTにIE850AのLPDクロックを追加  ex\_socunitinfo2 EX\_UNIT\_TRCSELの外部トレースにIE850Aを追加  ex\_socunitinfo2 EX\_UNIT\_WORKRAMにIE850Aを追加  3.2.3 メモリ操作のex\_memwrite,ex\_memfill,ex\_memcopyに新規エラーEX\_SE\_SECAREAACCERRを追加 | 新井 |  |  |
|  | 2017/08/28 | 誤記訂正  ・ex\_stop [Description]を修正 | 横山 |  |  |
|  | 2017/08/30 | RTOSタスクアナライザ対応  ・ex\_rh\_trctune [Arguments]、[デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2017/08/31 | G4世代MCU CPUコアデータアクセスイベントのエラー応答対応  ex\_rh\_evncond  ・[Return Values]にエラーコード(EX\_PE\_ACCEVNDATANOSPT)  を追加  ・[Description]-[引数説明]-stat-<CPUコアイベント条件>で  　記載していたエラーコード(EX\_PE\_DATA)を変更  ex\_rh\_getenv  ・[Arguments]-funcinfo[2].b2の説明追加  ・[Description]-[引数説明]-[funcinfoの各情報に対して  関連するEXEC I/F]-funcinfo[2].b2の説明追加 | 有鹿 |  |  |
| ↓ | 2017/08/31 | G4世代MCU TEU\_CPUの命令系範囲イベントデバイス不具合対応  ex\_rh\_evncond  ・[Return Values]にエラーコード(EX\_PE\_INSTEVNRANGNOSPT)  を追加  ・[動作説明]にエラー応答する説明を追加 | 有鹿 |  |  |
| ↓ | 2017/08/31 | G4世代E2M(製品チップ)対応  ex\_rh\_getenv  　・[Arguments]-funcinfo[12]～funcinfo[15]の説明を追加  ・[Description]-[引数説明]-[funcinfoの各情報に対して  関連するEXEC I/F]-funcinfo[12]～funcinfo[15]の説明追加  ex\_timeread2,ex\_rh\_timeread,ex\_timeevent2,ex\_rh\_timeevent  ex\_rh\_performancesw,ex\_rh\_performanceclear, ex\_rh\_performanceread,ex\_rh\_performanceset  　・[Return Values]にエラーコード追加  ・[動作説明]にエラー応答する説明を追加 | 有鹿 |  |  |
| ↓ | 2017/09/01 | ・ex\_setenv  IE850A外部トレース対応のため、H/Wに合わせてトレース記録サイズを追加。 | 畑原 |  |  |
| ↓ | 2017/09/04 | ・ex\_socunitinfo2  　表3-5にEX\_UNIT\_CMFIDとEX\_UNIT\_OPTIDを追加 | 永井 |  |  |
| ↓ | 2017/09/04 | G4世代外部ソフトトレース単線出力対応  1.の表1-2を修正  2.9 [対応コア世代]を追加  ・ex\_rh\_sfttrcrawmemread [Description]-[デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2017/09/08 | ・表1-16「 EXRH850G4.DLLファイルのプロパティ」を新規追加。  ・表1-17 「ファイルセット一覧」  に EXRH850G4.DLL,EXRH850G4.INIを新規追加。  ・表2-1～表2-4 「インターフェース関数一覧」に  G4のI/Fサポート情報を追記。  また、ex\_socunitinfo2 unit\_type別のサポート情報を追記。 | 畑原 |  |  |
| ↓ | 2017/09/13 | ・ex\_setenv trcmem\_size のEX\_M\_TRCMEM256M 説明を訂正。  正)256Mトレースメッセージアドレス(IE850A時のみ)  誤)256Mトレースメッセージアドレス(SPボード搭載/IE850A時のみ)  ・ex\_socunitinfo2 [Return Values]欄から  エラー「EX\_DE\_OCDIDCHKFAILED」の定義を削除。 | 畑原 |  |  |
| ↓ | 2017/09/19 | EXEC V4.01.00開発 機能設計完了。  ・文章番号を「IDF-16-007102」から「LLWEB- 10008657」に変更。 | 横山 | 畑原 | 佐々木 |
| ↓ | 2017/10/11 | EXEC V4.01.00開発 翻訳依頼 | 畑原 |  |  |
| ↓ | 2017/11/01 | EXEC V4.01.00開発 機能設計DR指摘の対応。  ・IE850Aの記載が適切でない部分の見直し  ・名称訂正。IECUBE2からIE850に変更。 | 横山 | 畑原 | 佐々木 |
| ↓ | 2017/11/6 | ・ex\_resetrun [Return Values]に以下を追加  　EX\_SE\_ASPENABLE、EX\_SE\_SFTTRACEENB  　また、「Description」に上記エラーの説明を追加 | 永井 |  |  |
| ↓ | 2017/11/9 | ・ex\_socunitinfo2 EX\_UNIT\_FID [デバッガ関係者向け注意事項]に(2)～(5)を追加 | 永井 | 畑原 | 佐々木 |
| ↓ | 2017/12/05 | EXEC V4.01.00開発 翻訳依頼時の指摘修正。  (本修正で仕様変更となる部分はない)  ・表1-2  未記載部分に必要情報を追記。  No.66　LPD端子の「備考」欄  No.3　 E2の「V.4.01」欄  No.53　強制ブレークの「V.4.01」欄  ・1.4.4  誤)そのため先にE2とE2 Lite合計で4台以上PCに接続されている場合は  正)そのため先にE2/E2 Lite/IE850A合計で4台以上PCに接続されている場合は  ・表 2-1 インターフェース関数一覧  カバレッジ関連の記載を削除  ・ex\_socunitinfo2  ・表2-1  誤)EX\_UNIT\_DFLNUMINFO データフラッシュブロック個数情報  正)EX\_UNIT\_DFLNUMINFO データフラッシュブロック個数情報取り出し  ・ex\_socunitinfo2  ・表2-1  誤)EX\_UNIT\_TRCSEL 使用するトレース出力タイプ  正)EX\_UNIT\_TRCSEL 使用するトレース出力タイプを設定/参照する  ・表2-2  誤)EX\_UNIT\_WORKRAM ワークRAM領域の開始アドレスを設定/参照  正)EX\_UNIT\_WORKRAM ワークRAM領域の開始アドレスを設定/参照  　　 ワークRAM領域のサイズを参照  ・ex\_initexec  誤)ホットプラグインアダプタにE1/E20/E2/IE850Aユーザインターフェースケーブルを未接続  正)ホットプラグインアダプタにE1/E20/E2ユーザインターフェースケーブルを未接続  ・ex\_rh\_getenv  　誤)パフォーマンス機能  正)パフォーマンス計測機能  ・ex\_socunitinfo2  EX\_UNIT\_OPTID　unit\_info[2]の説明訂正  誤)unit\_info[0]で指定したIDコード番号に対応したIDコード  正)unit\_info[0]で指定したIDコード番号に対応したIDコード1  ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_FID>  [引数説明]  不要な用語(,CustomerID)を削除  用語の変更(I/F⇒パラメータ)  　[デバッガ開発者向け注意事項]  　　用語の変更(解除しない⇒行わない)  ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_CMFID>  [引数説明]  　　表内の誤記を修正(m個中のn番目⇒m個中のm番目)  　[動作説明],[デバッガ関係者向け注意事項]  　　用語の変更(I/F⇒パラメータ)  誤記を修正(ex\_socunitinfo⇒ex\_socunitinfo2)  ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_OPTID>  　[引数説明]  　　(例2)の表の欄外の注記番号を修正(※1⇒※2)  ・1.2.1　表内のTEU\_CPU  ・ex\_rh\_getenv [Arguments]-mcuinfo[2] bit4  ・ex\_rh\_getenv [Arguments]-funcinfo[0] bit3  ・3.2.7 (1)トレース関数 [スレーブトレースの注意事項]の(3)  ・ex\_rh\_trceventsel2 [Arguments]-item bit[15:12], bit23  　用語の変更(/⇒,)  ・ex\_rh\_evncond 表3-9 extd,extd\_mask  TEUスレーブイベント条件欄を見せ消しに修正  ・表3-7、表3-10  　実線に修正  ・ex\_timeevent2 [デバッガ開発者向け注意事項]  　G4世代の注意事項を[動作説明]欄の最後に移動 | 畑原  横山  有鹿  永井 | 畑原 | 佐々木 |
| ↓ | 2017/12/19 | EXEC V4.01.00開発 翻訳依頼時の指摘修正。  ・表紙  RH850G3/G4コア用 E1/E20/E2/IE850A/IECUBE2IE850  　エミュレータF/W機能仕様書 の仕様書名を訂正。  ・表1-1 No.30 リニア電源供給機能  [EMU種別] E2->OCDに訂正  [備考] E1/E20エミュレータではサポートしない　を追記。  ・表1-2 No.53 強制ブレーク  EML種別欄を 「←」から「ALL」に訂正。  ・表1-2 No.66 トレース方式 LPD端子  EML種別欄を 「－」から「E2」に訂正。  ・表2-4  下記、定義が重複しているため削除。  ex\_timesw ex\_timerun ex\_timeclear  ex\_timeread ex\_timemode ex\_timeread2  ex\_timeevent2 ex\_timeevent3 ex\_rh\_timeread  ex\_rh\_timeevent ex\_rh\_performancesw  ex\_rh\_performanceclear ex\_rh\_performanceread  ex\_rh\_performanceset  ・ex\_rh\_evncond。誤記訂正。  正)ex\_rh\_evncond stat=  誤)ex\_rh\_evncond\_stat=  ・ex\_boardinfo2 [Arguments] 誤記訂正。  正)E1/E20/E2/IE850A  誤)E1/E20/E2/IE850  ・DataFlashセキュリティIDコード認証 用語統一  訂正後)エミュレータ起動  訂正前)EXEC起動  ・ex\_trcswの記載  英語では、排他関係(OR)を表すのにコンマよりはスラッシュを  使うので和文と英文を一致するために記載を訂正。  訂正後)<TCU\_GRM>,<TCU\_CRM/TCU\_GIO/TCU\_GVC>,<TCU\_KVC>,…  訂正前)TCU\_GRM,TCU\_CRM/TCU\_GIO/TCU\_GVC,TCU\_KVC,…  ・表3-7、表3-10の罫線  実線に統一して訂正。 | 畑原 |  |  |
| ↓ | 2017/12/25 | EXEC V4.01.00開発 翻訳依頼時の指摘修正。  ・表1-2 No.53 強制ブレーク  EML種別欄を 「←」から「ALL」に訂正。  ・表2-4  下記、定義が重複しているため削除。  ex\_timesw ex\_timerun ex\_timeclear  ex\_timeread ex\_timemode ex\_timeread2  ex\_timeevent2 ex\_timeevent3 ex\_rh\_timeread  ex\_rh\_timeevent ex\_rh\_performancesw  ex\_rh\_performanceclear ex\_rh\_performanceread  ex\_rh\_performanceset  ・ex\_trcswの記載  　英語では、排他関係(OR)を表すのにコンマよりはスラッシュを  使うので和文と英文を一致するために記載を訂正。 | 畑原 |  |  |
| ↓ | 2018/01/17 | EXEC V4.01.00開発 翻訳依頼時の指摘修正。  ・誤記修正  「SPボート」→「SPボード」　(9箇所)  「トーレス・データ」→「トレース・データ」　(2箇所)  補足)翻訳した英語版仕様書と同等 | 畑原 |  |  |
| 13 | 2018/1/17 | V4.01.00での修正履歴を削除 | 横山 |  |  |
| ↓ | 2018/1/17 | ・ex\_rh\_trctune  [Arguments]、[動作説明] <trc\_tunetype= EX\_TRC\_TRCRAMOPTION>を修正 | 横山 |  |  |
| ↓ | 2018/1/17 | 2.1.2 インタフェース関数一覧の表を修正  ・ex\_socunitinfo2  [Arguments]、[Return Value]を修正  <unit\_type=EX\_UNIT\_SLAVEIDSEL> [デバッガ開発者向け注意事項]を修正  [Description]に<unit\_type=EX\_UNIT\_CLUSTERSEL>を追加  unit\_type別呼び出し可否の表を修正 | 横山 |  |  |
| ↓ | 2018/1/18 | ・ex\_rh\_getenv  [Arguments]、[Description]-[引数説明]を修正  ・ex\_socunitinfo2  <unit\_type=EX\_UNIT\_COREPESEL> [デバッガ開発者向け注意事項]  <unit\_type=EX\_UNIT\_SLAVEIDSEL> [デバッガ開発者向け注意事項]  ・ex\_rh\_trctune  [Arguments]を修正  [Description]に<trc\_tunetype=EX\_TRC\_WPTSMSG>を追加  [Description]に<trc\_tunetype=EX\_TRC\_GBUFCADET>を追加  [Description]に<trc\_tunetype=EX\_TRC\_GETGTS>を追加 | 横山 |  |  |
| ↓ | 2018/1/26 | ・ex\_rh\_getenv  [Arguments]、[Description]-[引数説明]を修正  ・ex\_rh\_trctune  [Description]-<trc\_tunetype=EX\_TRC\_WPTSMSG>を修正  [Description]-<trc\_tunetype=EX\_TRC\_GBUFCADET>を修正 | 横山 |  |  |
| ↓ | 2018/1/31 | ・ex\_rh\_sfttrcrawmemread  [デバッガ開発者向け注意事項]を修正 |  |  |  |
| ↓ | 2018/02/08 | ・ex\_memwrite [引数説明] <カテゴリ別の説明>の2．  　コードフラッシュ関連領域以外の領域を指定した場合のエラー記載を修正(EX\_PE\_NOFLASHAREA⇒EX\_PE\_DATA) | 有鹿 |  |  |
| ↓ | 2018/2/19 | ・ex\_rh\_evncond  [Description]stat引数<TEUスレーブイベント条件>にクラスタ番号指定の内容を追記  [デバッガ開発者向け注意事項]にクラスタ番号指定の内容を追記  ・ex\_rh\_evnintgcond  [デバッガ開発者向け注意事項]にクラスタ番号指定の内容を追記 | 新井 |  |  |
| ↓ | 2018/2/22 | ・イベント条件  複数CRAMに対応するため、TEUのスレーブイベント番号で使用できない番号を7fe0h-7fffhに変更  それに伴い、ex\_rh\_evncondのno引数およびex\_rh\_evnintgcondのevent引数で使用できるTEUスレーブイベント番号を4000h-7fdfhに変更 | 新井 |  |  |
| ↓ | 2018/3/1 | ・表1-1 G3世代のV4.02のサポート機能を記載  ・表1-2 G4世代のV1.00のサポート機能を記載  ・表1-15、表1-16のバージョンを修正 | 永井 |  |  |
| ↓ | 2018/3/2 | ・2.10 新規追加  ・ex\_exit [動作説明]にCANブートデバッグの説明を追加  ・ex\_reset [Return Values]にエラーを追加、[動作説明]にCANブートデバッグの説明を追加  ・ex\_socunitinfo2 [Arguments]、[動作説明]にCANブートデバッグの説明を追加  ・ex\_resetrun [Return Values]にエラーを追加、[動作説明]にCANブートデバッグの説明を追加 | 永井 |  |  |
| ↓ | 2018/3/13 | ・ex\_socunitinfo2 <unit\_type=EX\_UNIT\_LPDOPT>  unit\_info[1]の説明にR-Car ICUMXAでは設定値を無視する仕様を  　補足として追記。 | 島田 |  |  |
| ↓ | 2018/3/14 | 2.9 [対応コア世代]を修正 | 横山 |  |  |
| ↓ | 2018/3/14 | G4世代メモリアクセス仕様変更に伴う対応  ・ex\_memread,ex\_memfill,ex\_memcopy  [Return Value]：エラーコードを新規追加  [動作説明]：メモリアクセス仕様変更に伴う説明追加  ・ex\_memwrite,ex\_run,ex\_resetrun,ex\_step  [Return Value]：エラーコードを新規追加  既存エラーコード(EX\_DE\_SECAREAACCERR)の説明文の修正  [動作説明]：メモリアクセス仕様変更に伴う説明追加 | 有鹿 |  |  |
| ↓ | 2018/4/03 | EXEC V1.00.00開発 機能設計完了。 | 有鹿 | 畑原 | 佐々木 |
| ↓ | 2018/4/26 | EXEC V1.00.00開発 単体テスト完了。  誤記訂正  ・ex\_rh\_getenv [Argument]を修正 | 横山 | 畑原 | 佐々木 |
| ↓ | 2018/06/28 | EXEC V1.00.00開発 翻訳指摘対応 | 新井 |  |  |
| 14 | 2018/7/4 | V4.02.00での修正履歴を削除 | 横山 |  |  |
| ↓ | 2018/3/14 | IE850Aの外部トレース対応  ・表紙を修正  ・1.4.7 ファイルセット一覧の表を修正  ・2.2.1 起動 を修正  ・ex\_initexec [Arguments]、[Return Values]、[引数説明]、[動作説明]を修正  ・ex\_extpower [Arguments]、[Return Values]、[引数説明]、[動作説明]を修正  ・ex\_boardinfo2 [Format]、[Return Values]、[Arguments]、[引数説明]、[動作説明]を修正  ・ex\_setenv [Description]を修正  ・ex\_socunitinfo2 [Arguments]、[Return Values]を修正  <unit\_type=EX\_UNIT\_TRCSEL> [引数説明]、[動作説明]を修正  <unit\_type=EX\_UNIT\_EXTTRACERATE>を追加  表3-5を修正 | 横山 |  |  |
| ↓ | 2018/3/29 | 誤記訂正  ・ex\_rh\_getenv [Arguments]を修正  IE850Aの外部トレース対応  ・表紙を修正  ・ex\_initexec [Return Values]、[動作説明]を修正  ・ex\_socunitinfo2 [Arguments]、[Return Values]を修正  <unit\_type=EX\_UNIT\_TRCSEL> [動作説明]を修正  <unit\_type=EX\_UNIT\_EXTTRACERATE> [動作説明]を修正  ・ex\_trcrawmeminfo [Return Values]、[動作説明]を修正  ・ex\_trcrawmemread [Return Values]、[動作説明]を修正 | 横山 |  |  |
| ↓ | 2018/4/26 | IE850Aの外部トレース対応  ・ex\_initexec [Return Values]、[動作説明]を修正  ・ex\_socunitinfo2 [Return Values]を修正  <unit\_type=EX\_UNIT\_RSU> [デバッガ関係者向け注意事項]を修正  <unit\_type=EX\_UNIT\_TRCSEL> [動作説明]を修正  <unit\_type=EX\_UNIT\_EXTTRACERATE> [動作説明]を修正 | 横山 |  |  |
| ↓ | 2018/6/29 | IE850Aの外部電源供給機能の仕様変更  ・ex\_extpower [Arguments]、[Return Values]、[引数説明]、[動作説明]を修正  ・ex\_initexec [Arguments]、[引数説明]、[動作説明]を修正  ・ex\_exit [動作説明]を修正 | 横山 |  |  |
| ↓ | 2018/7/6 | ・U2A  2.1.2 インターフェース関数一覧 表2-2 ex\_socunitinfo2 EX\_UNIT\_SVRを追加  2.2.1 起動のG4世代MCU起動手順にex\_socunitinfo2 EX\_UNIT\_SVRを追加  3.2.2 ハードウェア条件設定のex\_getstatusにDEEP STOPモード, Cyclic RUNモード, Cyclic STOPモードの通知ビットを追加  3.2.2 ハードウェア条件設定 ex\_socunitinfo2のEX\_UNIT\_FLFUNCINFOでサポートマクロにFL\_RV28F\_STDを追加  3.2.2 ハードウェア条件設定のex\_socunitinfo2にEX\_UNIT\_SVRを追加  3.2.2 ハードウェア条件設定のex\_socunitinfo2 unit\_type=EX\_UNIT\_LPDOPTにマップモード指定引数のunit\_info[8]を追加  3.2.2 ハードウェア条件設定のex\_socunitinfo2に新規エラーEX\_PE\_MAPMODENOMATCH,EX\_DE\_SINGLEMAPONLYを追加  3.2.3 メモリ操作 ex\_memwriteにU2Aのフラッシュエキストラにライトする場合の引数, 動作に関して記載  3.2.8 ハードウェア直接操作用関数にex\_recoveryextraflashを追加  ・IE850A  3.2.8 ハードウェア直接操作用関数にex\_setselid, ex\_getselid, ex\_setnid, ex\_getnidを追加 | 新井 |  |  |
| ↓ | 2018/7/9 | ・PDR指摘修正  2.1.2 インターフェース関数一覧にex\_recoveryextraflashを追加  2.2.1 起動でホットプラグインではEX\_UNIT\_SVRが不要なため削除  3.2.2 ハードウェア条件設定のex\_socunitinfo2 unit\_type=EX\_UNIT\_RSUに指定マップモードとデバイスマップモードが異なる場合EX\_PE\_MAPMODENOMATCHエラーになる説明を追加  3.2.2 ハードウェア条件設定のex\_socunitinfo2 unit\_type=EX\_UNIT\_SVRにDFから要否を取得する説明を追加  EX\_UNIT\_RSUより前に発行するように修正  3.2.3 ex\_socunitinfo2にex\_recoveryextraflashを移動  EX\_UNIT\_EFRECOVERYとし、全コアブレーク中のみ発行可能に修正  4.2 インターフェース仕様にスタンバイモードの項目を追加 | 新井 |  |  |
| ↓ | 2018/7/11 | IE850A対応  ・2.7非同期デバッグに関する動作仕様の[非同期デバッグモードにおける制限事項]を修正  ・ex\_getstatus [Format]、[引数説明]、[デバッガ開発者向け注意事項]を修正  ・ex\_trcmode [デバッガ開発者向け注意事項]、[非同期デバッグモード時の注意事項]を修正  ・ex\_trcdelay [デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2018/7/24 | IE850A対応  ・2.7非同期デバッグに関する動作仕様の[非同期デバッグモードにおける制限事項]を修正(7/11の修正を元に戻す)  ・ex\_getstatus [Argument]、[引数説明]を修正  ・ex\_trcdelay [引数説明]、[デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2018/7/31 | ・IE850A  3.2.1 ex\_extpowerのエラーにEX\_FE\_FANERRORとEX\_FE\_VOUTERRORを追加  3.2.2 ex\_getstatusのエラーにEX\_FE\_FANERRORとEX\_FE\_VOUTERRORを追加  4.2 インターフェース仕様のEX\_WM\_ERRSTATにIE850Aのエラーを追加 | 新井 |  |  |
| ↓ | 2018/8/23 | IE850A対応  ・ex\_rh\_trctune [Arguments]、[デバッガ開発者向け注意事項]を修正 | 横山 |  |  |
| ↓ | 2018/9/7 | IE850A対応  ・3.2.1 ex\_extpowerとex\_getstatusに、新規仕様を追加  IE850AのVCCLEDが点滅中は電源供給を停止し、電源供給機能も設定変更できない  ・3.2.1 ex\_extpower [Description]を修正  VOUT1からは3.3Vしか供給できないが、供給電圧値は1.2V～5.5V範囲内の値を指定すること。  ・3.2.8 ex\_setselid, ex\_getselid, ex\_setnid, ex\_getnidの[Description]に追加  RH850G4用EXECのみサポートであること  EXEC V1.01.00開発 機能設計完了。 | 新井 | 畑原 | 佐々木 |
| ↓ | 2018/9/11 | ・表1-1 V4.03の列を追加  ・表1-2 V1.01の列を追加  ・表1-11、表1-12、表1-13 Vistaを削除  ・図1-2 タイトルにIE850Aを追加  ・表1-15、表1-16 モジュールのバージョン表示を修正 | 永井 |  |  |
| ↓ | 2018/9/20 | EXEC V1.01.00開発 詳細設計完了 | 新井 | 畑原 | 佐々木 |
| ↓ | 2018/9/21 | ・ex\_socunitinfo2にエラーEX\_FE\_EFRECOVERYFAILEDを追加  ・EX\_UNIT\_EFRECOVERYの説明にEX\_FE\_EFRECOVERYFAILEDエラーを追加 | 新井 |  |  |
| ↓ | 2018/10/4 | EXEC V1.01.00開発 製造完了 | 新井 | 畑原 | 佐々木 |
| 15 | 2018/9/12 | ・2.6仮想化支援機能に関する動作仕様にRH850 G4の記載を追加  ・ex\_getstatus cpustat.cs\_vm\_gpidを追加  　[引数説明]：cpustatにG3、G4の説明を追加  ・ex\_socunitinfo2  　<unit\_type=EX\_UNIT\_HEGESEL>を追加  　[Description]<unit\_type=EX\_UNIT\_COREPESEL>にG4の説明を追加  　表3-5にEX\_UNIT\_HEGESELを追加  ・3.2.7 アナライザ (1)トレース関数にex\_rh\_trcfilterを追加  ・ex\_rh\_sfttrcfilterのfilter\_typeにEX\_FILTERTYPE\_SPIDを追加 | TSSR児玉 |  |  |
| ↓ | 2018/9/12 | ・「三層」を「仮想マシン」に文言修正(全体)  ・2.6 RH850 G4の記載を追加  ・ex\_rh\_getenv funcinfo[0]にSPIDフィルタリング情報の追加  ・ex\_getstatus es\_brombreakの記述を削除  ・ex\_regread デバッグ対象外のCPU動作モード時の動作説明を追加 [仮想マシン対応時の注意点(G4)]を追加  ・ex\_regwrite デバッグ対象外のCPU動作モード時の動作説明を追加 [仮想マシン対応時の注意点(G4)]を追加  ・ex\_rh\_evncond [仮想マシン対応時の注意点(G4)]を追加 phase2に関する説明を削除  ・3.2.7 (1) [仮想マシン対応時の注意点(G4)]を追加  ・ex\_rh\_trceventsel2 itemのbit2にCPU動作モード切り替え情報のトレース・データ取得有無を追加  ・3.2.7 (2) [仮想マシン対応時の注意点(G3)]と[仮想マシン対応時の注意点(G4)]を追加  ・3.2.7 (4) [仮想マシン対応時の注意点(G4)]を追加  ・ex\_rh\_sfttrcmode VMに関する設定はG3世代のみを追記  ・ex\_rh\_performanceset 表3－22に計測項目を追加  ・4.1 仮想マシンステータスのコールバックタイミングを記載  ・4.2 仮想マシンステータス情報のコールバックについて記載 EX\_M\_BROMAREA\_BKの記述を削除 | 永井 |  |  |
| ↓ | 2018/9/26 | ・2.6 ハードウェア条件設定機能にステップ実行のオプション説明を追加  ・ex\_getstatus exestatのbit14にes\_vm\_timeoutを追加  ・socunitinfo2 EX\_UNIT\_HEGESELのunit\_info[0].bit30にステップ実行のオプションをアサイン [動作説明]ステップ機能オプションの説明を追加  ・ex\_step [Description]にステップ機能オプションの説明を追加 [仮想マシン対応時の注意点(G4)]を追加 | 永井 |  |  |
| ↓ | 2018/10/4 | ・ex\_dcuregread、ex\_dcuregwrite codeの指定方法を追記 | 永井 |  |  |
| ↓ | 2018/11/07 | E2UH/E2H/E2GUH/E2GH対応  ・1章 表1-2にE2UH/E2H/E2GUH/E2GH WS対応時のバージョン欄を追加  ・3.2.2　ex\_socunitinfo2 [Arguments] [Description] [Return Values]  unit\_type=EX\_UNIT\_LPDOPT unit\_info[8]に指定値を新規追加  　注記にエラー応答の説明を追加  　unit\_type=EX\_UNIT\_LPDOPT unit\_info[8]の動作説明に新規マップモードの記載を追加  　unit\_type=EX\_UNIT\_RSUの動作説明に新規マップモードの記載を追加 | 有鹿 |  |  |
| ↓ | 2018/11/16 | ・ex\_regread、ex\_regwrite [動作説明]、[仮想マシン対応時の注意点(G4世代)]にエラー応答の対象外となるレジスタの説明を追加 | 永井 |  |  |
| ↓ | 2019/1/8 | ・ex\_initexec  　[Return Values]にエラーコードを追加  　[動作説明]に追加したエラーコードの説明を追加  ・ex\_socunitinfo2  　[Return Values]にエラーコードを追加  　EX\_UNIT\_RSUの[動作説明]に追加したエラーコードの説明を追加 | 永井 |  |  |
| ↓ | 2019/1/17 | ・1.4.4 EXECの64bitアプリ対応について説明を追加  ・1.4.5 EXECの64bit版を追加  ・1.4.6 EXECの64bit版の表示例を追加  　　　　今回開発のバージョンに修正  ・1.4.7 EXECの64bit版のファイルを追加 | 永井 |  |  |
| ↓ | 2019/02/04 | [初期停止・スタンバイモード時のデバッグ機能対応]  ・2.1.2　表2.2(2/4, 3/4)  新規I/Fの記載追加および表の修正  ・2.2.1　起動  シーケンスに新規I/Fの呼び出しを追加および手順番号を修正  ・2.4  低消費電力モード時の動作仕様　Cyclicモード中の動作仕様を追加  ・2.7　非同期デバッグに関する動作仕様  　初期停止・スタンバイモード優先デバッグ環境モード有効時の注意事項を追加  ・2.11  初期停止・スタンバイモード優先デバッグ環境モードに関する動作仕様の章を新規追加  ・3.2 関数詳細説明  以下のI/F関数の章に、初期停止・スタンバイモード優先デバッグ環境モード有効時のパラメータ、戻り値、動作説明、注意事項の記載を追加  ex\_reset、ex\_getstatus、ex\_socunitinfo2、ex\_memread、  ex\_memwrite、ex\_memfill、ex\_memcopy、ex\_regread、  ex\_regwrite、ex\_rh\_evncond、ex\_run、ex\_step、ex\_stop、  ex\_hwbrkevent、ex\_swbrkcond  ・4.　非同期通信インターフェース仕様  初期停止・スタンバイモード優先デバッグ環境モード有効時のCPUステータスに関する記載を追加 | 有鹿 |  |  |
| ↓ | 2019/02/25 | [初期停止・スタンバイモード時のデバッグ機能対応]  ・3.2 関数詳細説明　ex\_regread  　初期停止・スタンバイモード優先デバッグ環境モード有効時の注意事項に、ダミー値を返すレジスタの説明を追加 | 有鹿 |  |  |
| ↓ | 2019/2/27 | ・ex\_initexec  　[引数説明]にIE850Aでの動作を追記  　[動作説明]にIE850Aでの動作仕様を追記  ・ex\_extpower  　[Return Values]にエラーを追加  　[動作説明]にIE850Aでの動作仕様を追記  　[デバッガ開発者向け注意事項]にIE850Aでの注意を追記  ・ex\_getstatus  　[Return Values]にエラーを追加  　[動作説明]にIE850Aでの動作仕様を追加  　[デバッガ開発者向け注意事項]にIE850Aでの注意を追記  ・ex\_boardinfo2  　[Return Values]にエラーを追加  　[動作説明]にIE850Aでの動作仕様を追加  　[デバッガ開発者向け注意事項]にIE850Aでの注意を追記  ・非同期通信インターフェース仕様  　<status\_type=EX\_WM\_ERRSTAT指定時>にbit4を追加  　[デバッガ開発者向け注意事項]にIE850Aでの注意を追記 | 新井 |  |  |
| ↓ | 2019/03/07 | ・2.2.1 起動にデバッガ開発者向け注意事項を追加  ・ex\_extpower  　[Return Values]のエラー説明文を修正  　[デバッガ開発者向け注意事項]にIE850Aでの注意を修正  ・ex\_getstatus  　[Return Values]のエラー説明文を修正  　[デバッガ開発者向け注意事項]にIE850Aでの注意を修正  ・ex\_boardinfo2  　[Return Values]のエラー説明文を修正  　[デバッガ開発者向け注意事項]にIE850Aでの注意を修正  ・非同期通信インターフェース仕様  　<status\_type=EX\_WM\_ERRSTAT指定時>のエラー説明を修正  　[デバッガ開発者向け注意事項]にIE850Aでの注意を修正 | 新井 |  |  |
| ↓ | 2019/3/11 | ・表1-1、表1-2のサポート機能一覧を更新 | 永井 |  |  |
| ↓ | 2019/03/11 | ・表3-19 IE850Aの無効メッセージの説明を追記。  RH850 EXEC V1.02.00開発 機能設計完了。 | APIP3  永井  2019/3/11 | APIP3  畑原  2019/3/11 |  |
| ↓ | 2019/03/19 | [初期停止・スタンバイモード時のデバッグ機能対応]  Cyclic STOP/DEEP STOP/ STOP状態中に操作できるデバッグ機能を  強制リセットのみとし、デバイス動作に合わせてデバッグ機能の  制限を増やした。  ・全体  　用語変更(初期停止・スタンバイモード優先デバッグ環境モード↓初期停止・スタンバイモードデバッグ環境)  ・2.4　(3),(4),(7)  　低消費電力モード時の動作仕様の説明を修正(対象モード中はエラー応答する旨の記載)  ・2.11  　本文中の既存デバッグ環境との違いの説明文を修正  　表2-11～2-14の記載を修正  　[制限事項]のホットプラグイン接続の説明を修正  ・3.2 関数詳細説明  以下のI/F関数の章に、初期停止・スタンバイモードデバッグ環境が有効時の戻り値、注意事項の記載を追加・修正  　ex\_pincont, ex\_memread, ex\_memwrite, ex\_memfill,  ex\_regread, ex\_regwrite, ex\_rh\_evncond, ex\_rh\_evnintgcond,  ex\_stop, ex\_hwbrkevnent, ex\_trcrun, ex\_trcmode,  ex\_trcclear, ex\_trcevent, ex\_trcdelay, ex\_rh\_trceventsel2,  ex\_trcbufcont, ex\_trcrawmeminfo, ex\_trcrawmemread,  ex\_rh\_trcfilter, ex\_timeread2, ex\_dcuregread,  ex\_dcuregwrite  ・3.2.7 (2)外部ソフトトレース単線出力関数  　初期停止・スタンバイモードデバッグ環境が有効時の注意事項の記載追加 | 有鹿 |  |  |
| ↓ | 2019/03/20 | RH850 EXEC V1.02.00開発 機能設計完了。 | APIP3  永井  2019/3/20 | APIP3  畑原  2019/3/20 |  |
| ↓ | 2019/03/27 | EX\_UNIT\_EFRECOVERY  正式名称となった「SWASリカバーコマンド」に言葉を統一  P.27, P.138, P152, P.188 | 新井 |  |  |
| ↓ | 2019/03/28 | [初期停止・スタンバイモード時のデバッグ機能対応]  非同期デバッグモードは未サポートの機能制限を追加  ※上記修正に加えて、以前追加した記載の文言を修正  2.11章　[機能概要]  ・従来との違いを示した記載を修正  　・表2-11，2-12の制限内容を修正  　・表2-13，2-14を削除  3章  　・以下のI/Fの章の「初期停止・スタンバイモードデバッグ環境が有効時の注意事項」の記載を修正、追加  　ex\_reset, ex\_getstatus, ex\_socunitinfo2(EX\_UNIT\_TRCSEL, EX\_UNIT\_ASYNCPEOPT, EX\_UNIT\_HEGESEL), ex\_run, ex\_stop  ・ex\_socunitinfo2の戻り値に記載を追加(EX\_SE\_CPUSTOP\_NOSPRT, EX\_SE\_STANDBY\_NOSPRT) | 有鹿 |  |  |
| ↓ | 2019/03/29 | RH850 EXEC V1.02.00開発 機能設計完了。 | APIP3  永井  2019/3/29 | APIP3  畑原  2019/3/29 |  |
| ↓ | 2019/04/19 | [初期停止・スタンバイモード時のデバッグ機能対応]  ICUM有効でかつCyclicRunモード時の操作を制限する対応を追加  2.11章　[機能概要]  ・CyclicRun中・CyclicRunに伴うCyclic Disable中のコアステータス状態についての記載に、ただし書きの文言を追加  ・表2-11に※10の記載追加  ・表2-12に※6の記載追加  3章  ・以下のI/Fの章の「初期停止・スタンバイモードデバッグ環境が有効時の注意事項」の記載を修正、追加  ex\_pincont, ex\_socunitinfo2(EX\_UNIT\_TRCSEL), ex\_memread,  ex\_memwrite, ex\_memfill, ex\_regread, ex\_regwrite,  ex\_rh\_evncond, ex\_rh\_evnintgcond, ex\_stop, ex\_hwbrkevent,  ex\_trcrun, ex\_trcmode, ex\_trcclear, ex\_trcevent,  ex\_trcdelay, ex\_rh\_trceventsel2, ex\_rh\_trcbufcont,  ex\_trcrawmeminfo, ex\_trcrawmemread, ex\_rh\_trcfilter,  ex\_timeread2, ex\_dcuregread, ex\_dcuregwrite | 有鹿 |  |  |
| ↓ | 2019/04/25 | 初期停止・スタンバイモード時のデバッグ機能対応]  ホットプラグイン接続かつスタンバイ状態で強制リセットコマンドが発行された場合に制限する対応を追加  2.11章　表2-12　※7の記載追加  3章　ex\_reset  ・[Return Values]欄にエラーコードの記載を追加  ・初期停止・スタンバイモードデバッグ環境が有効時の注意事項」にエラー応答する記載を追加 | 有鹿  APIP3  永井  2019/5/10 | APIP3  　 畑原  2019/5/10 |  |
| ↓ | 2019/05/13 | 翻訳Grからの指摘事項対応  2.2.1章  ＜G3世代MCU起動時に必要なEXEC関数の手順(通常起動)＞  　　手順(11)、手順(12)の次の行の説明を修正  　＜G3世代MCUホットプラグイン起動の手順(OCDのみ)＞  　　手順(13)の次の行の説明を修正  　＜G4世代MCU 起動時に必要なEXEC関数の手順(通常起動)＞  　　手順(13)、手順(14)の次の行の説明を修正  　＜G4世代MCU ホットプラグイン起動の手順(OCDのみ)＞  　　手順(14)の次の行の説明を修正  2.11章  　[デバッガ関係者向け注意事項]  　　用語を修正(有効情報→有効にする値)  　表2-11 トレース行の初期停止欄  　　空欄だったため○を記載  　[主な制限事項] 【ホットプラグイン接続】  　　用語を修正((G3K(H))→G3KコアまたはG3K-Hコア)  　章全体  　　CyclicRun中→  CyclicRun中のメインコア  　　 メインコアがCyclicRun  CyclicRunに伴うCyclicDisableのコア→  　 CyclicDisable中のメインコア以外のコア  　 メインコア以外のコアがCyclicDisable  3.2章  ex\_getstatus  [Arguments] cpustat bit12の説明修正  　　表3-4 exestat bit13をbit14に修正してbit名、各実行状態の  　　記載を修正  　ex\_socunitinfo2 [Retrun Values]  　　EX\_FE\_WRITERENTRYERROR、EX\_FE\_WRITERCLOCKERRORの説明文の  　　間違い修正  　ex\_trcrawmemread 表3-19  　　No.3の備考欄の誤記修正  　ex\_dcuregread [Description] [codeの指定方法]  bit15の説明を修正  4.2章  　[デバッガ開発者向け注意事項]　(10)  　　例に記載している内容の一部削除  改訂履歴  　2018/11/07　対応デバイス名の修正  　2019/2/27　 ex\_initexecの修正内容の記載を修正  用語統一  HALT状態→HALTモード  STOP状態→STOPモード  DEEP STOP状態→DEEP STOPモード  Cyclic RUN状態→Cyclic RUNモード  Cyclic STOP状態→Cyclic STOPモード  ICU-M→ICUM | 有鹿  APIP3  永井  2019/5/13 | APIP3  　 畑原  2019/5/13 |  |