# カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジ が合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社 名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い 申し上げます。

ルネサスエレクトロニクス ホームページ (http://www.renesas.com)

2010年4月1日 ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社(http://www.renesas.com)

【問い合わせ先】http://japan.renesas.com/inquiry

#### ご注意書き

- 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
- 2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的 財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の 特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 3. 当社製品を改造、改変、複製等しないでください。
- 4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
- 5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところに より必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の 目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外 の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
- 6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
- 7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、 各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確 認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当 社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図 されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、意図 されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、 「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または 第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、デ ータ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
  - 標準水準: コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、 産業用ロボット
  - 高品質水準:輸送機器(自動車、電車、船舶等)、交通用信号機器、防災・防犯装置、各種安全装置、生命 維持を目的として設計されていない医療機器(厚生労働省定義の管理医療機器に相当)
  - 特定水準: 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器(生命維持装置、人体に埋め込み使用するもの、治療行為(患部切り出し等)を行うもの、その他 直接人命に影響を与えるもの)(厚生労働省定義の高度管理医療機器に相当)またはシステム 等
- 8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
- 9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
- 10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用 に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、 かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し て、当社は、一切その責任を負いません。
- 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお 断りいたします。
- 12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご 照会ください。
- 注1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレク トロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいい ます。



# SH7080 シリーズ用 E200F エミュレータ

# パフォーマンス測定方法

## 要旨

SH7080 シリーズ用フルスペックエミュレータ E200F は、ユーザプログラムの実行効率を測定できるパフォーマンス 解析機能を搭載しています。E200F エミュレータでは、オンチップパフォーマンス測定、AUD パフォーマンス測定の 2 種類のパフォーマンス解析機能を持っています。本書では、E200F エミュレータを用いてパフォーマンス解析機能の 使い方を説明します。

本書の使用例では、E200F エミュレータを単体で使用して説明しています。本書の内容は、異なるターゲットマイコンであっても E200F エミュレータであれば共通で利用することが可能です。

# 目次

1.	仕様	2
2.	使用機能説明	2
3.	ソフトウェアの準備	2
3.1	はじめに	2
3.2	E200Fエミュレータ付属品ソフトウェアのインストール	2
3.3	その他の必要なソフトウェアのインストール	2
4.	動作説明	3
4.1	High-performance Embedded Workshopの起動	3
4.2	ワークスペースを開く	4
4.3	イベントポイントによるパフォーマンス測定	. 10
4.4	データ条件(イベントポイント)によるパフォーマンス測定	. 23
4.5	実行命令数を使ったパフォーマンス測定の応用例	. 25
5.	よくある質問	. 27
5.1	パフォーマンス測定を行う上で注意する点はありますか?	. 27
6.	関連ドキュメント	28



#### 1. 仕様

E200Fエミュレータに搭載されているオンチップパフォーマンス測定機能は、デバイス内蔵カウンタにより、指定条件成立時から指定条件成立時までに要したサイクル数などを測定する機能です。サポートデバイスによって、サイクル数だけでなく、キャッシュミスの回数や、TLBミスの回数など、いろいろな項目を測定することもできます。なお、オンチップパフォーマンス測定機能とプロファイル機能は、同時に使用できません。

AUDパフォーマンス測定機能は、AUDイベント検出システムで指定されたイベント間の実行時間、実行回数などを 測定する機能です。タイマの分解能は20ns、100ns、400ns、1.6  $\mu$  sで測定可能な最大時間は、分解能20nsで約6時 間、分解能1.6  $\mu$  sで約20日です。

#### 2. 使用機能説明

本書では、E200F エミュレータ付属の CD-ROM、またはルネサス Web サイトからのダウンロードパッケージに含まれるサンプルプログラムを使用してパフォーマンス解析機能の使い方を説明します。

使用するツールのバージョンは次のとおりです。

エミュレータソフトウェア

E200F エミュレータデバッガ Version 2.04.00

#### 3. ソフトウェアの準備

3.1 はじめに

E200F エミュレータに付属する CD-ROM のソフトウェアをインストールします。

これにより本書で使用するサンプルプログラム(tutorial ワークスペース)がパソコン上に展開されます。

High-performance Embedded Workshop をインストール済みのパソコン上に E200F エミュレータに付属する CD-ROM のソフトウェアをインストールすることは可能です。この場合、インストール作業中に一部のダイアログ表示 が省略されることがあります。

#### 3.2 E200F エミュレータ付属品ソフトウェアのインストール

E200F エミュレータに付属する CD-ROM 内の HewInstMan.exe を実行してください。

インストールの詳細につきましては、ルネサス Web サイトに掲載の E200F エミュレータ用ユーザーズマニュアルを 参照していただき、インストール作業中は画面の指示に従って操作をしてください。ここではインストール手順は省略 します。

#### 3.3 その他の必要なソフトウェアのインストール

(1) ソフトウェアのインストールの際に、オートアップデートユーティリティを選択した場合、インターネット経由で各 ツールの最新版の存在を確認することが可能になります。



### 4. 動作説明

本章では、High-performance Embedded Workshop(HEW)を起動してパフォーマンス解析機能の使い方について 説明します。手順としては以下のようになります。



図 4.1 サンプルプログラムの実行手順

#### 4.1 High-performance Embedded Workshop の起動

まず、始めにユーザシステムを接続したE200FエミュレータとホストコンピュータをUSBケーブルで接続し、デバッグ 操作が可能であることを確認してください。

次にHigh-performance Embedded Workshopを起動します。

[スタート]メニューの[すべてのプログラム]から[Renesas]→[High-performance Embedded Workshop]→

[High-performance Embedded Workshop]で起動できます。





# 4.2 ワークスペースを開く

(1) High-performance Embedded Workshop 上に[ようこそ!]ダイアログボックスが表示されます。



[ようこそ!]ダイアログボックス内の[別のプロジェクトワークスペースを参照する]ラジオボタンを選択して[OK]ボタンを押してください。





(2) [ワークスペースを開く]ダイアログボックスが表示されます。

🛑 ワークスペースを	: 聞K				? 🗙
ファイルの場所型:	🗁 Tutorial_SH7080Series	•	🗢 🔁 (	* 🎫	
debug_e200f_sy	rstem				
Tutorial.hws					
ファイル名( <u>N</u> ):	Tutorial.hws			選折	र
ファイルの種類(工):	HEW Workspaces (*.hws)		•	キャンオ	211

本製品の CD-ROM のインストールが完了している場合、標準では次のフォルダ位置にワークスペー

- ス "Tutorial.hws" が格納されています。フォルダ位置を確認しながら順番に指定してください。ワークスペー
- ス "Tutorial.hws" が見つかりましたら指定し[選択]ボタンを押してください。

$C: \verb"FWorkSpace" Tutorial" E200F \verb"FSH7080Series" Tutorial\_SH7080Series" Tutorial.hws$
C:¥WorkSpace
<sup>L</sup> Tutorial
LE200F
└SH7080Series
<sup>L</sup> Tutorial_SH7080Series
└Tutorial.hws

【注】 ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その 場合は以下のディレクトリを指定してください。

<High-performance Embedded Workshop インストール先ディレクトリ>

ディレクトリ例:

C:\hew3\Tools\Renesas\DebugComp\Platform\E200F\SH7080Series\Tutorial\_SH7080Series

C:\hew2\Tools\Renesas\DebugComp\Platform\E200F\SH7080Series\Tutorial\_SH7080Series

(3) ワークスペースのバージョンが古い場合は次のダイアログボックスが表示されます。新しいバージョンにアップ デートするために[OK]ボタンを押してください。

High-pe	erformance Embedded Workshop 🛛 🔀
⚠	このワークスペースは以前のバージョンのHEWで作成されています。ワークスペースを開くとワークスペース、プロジェクトおよびセッションに関する情報がアップデー トされ、以前のバージョンのHEWでは開けなくなります。(元のファイルは "old version xxx"+ "元のファイル名" で保存されています。以前のバージョンのHEWで ワークスペースを開くためにはこれらのファイルを元のファイル名に戻してから開いてください。) このワークスペースを開きますか?
	<b>〇〇</b> キャンセル



(4) [ウォーニング]ダイアログボックスが表示されます。ディレクトリが移動されたワークスペースを開くため[はい]ボ タンを押してください。

ウォーニ	ンガ 🔀
♪	このワークスペースのディレクトリが移動されました。 旧ディレクトリ: C:¥Hew3¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥SH7080Series¥Tutorial 新ディレクトリ: C:¥WorkSpace¥Tutorial¥E200F¥SH7080Series¥Tutorial_SH7080Series ワークスペースを聞きますか?
	□ 今後表示しない(D) (はい(Y)) いいえ(N) キャンセル

(5) [本システムにないツールチェインのバージョン]ダイアログボックスが表示される場合は、プロジェクト名を選択して[OK]ボタンを押してください。

本システムにないツールチェインのバージョン 🛛 🛛 🔀
以下のプロジェクトの'Renesas SuperH RISC engine Standard Toolchain'ツールチェインバージョン '9.0.0.0' がありません。 アップグレードするプロジェクトを選択してください。
Tutorial

(6) [ツールチェインのバージョンの変更]ダイアログボックスが表示される場合は、利用するツールチェインバージョ ンを選択して[OK]ボタンを押してください。

ツールチェインのバージ	ョンの変更		? 🛛
ツールチェイン名: 現在のバージョン:	Renesas SuperH RISC engine 9.0.0.0	Standard	ОК <i><b>4</b>77211</i>
CPU種別( <u>C</u> ): ツールチェイン( <u>T</u> ):	SuperH RISC engine Renesas SuperH RISC engine	▼ Standard Toolc ▼	
ツールチェインバージョン	<b>1</b> 9.1.0.0		
Build phase OptLinker SH Assembler SH C/C++ Compiler SH C/C++ Library G	Version 9.02.00 7.01.01 9.01.00 enerator 3.00.02	<u>, , , , , , , , , , , , , , , , , , , </u>	"情≆服①



(7) [Summary]ダイアログボックスが表示される場合は、そのまま[OK]ボタンを押してください。

CI	nange Toolchain Version Summary	
	Summary :	
	Project name : Tutorial	~
	Renesas SuperH RISC engine Standard Toolchain was upgraded 9.0.0> 9.1.0.0.	
	[Standard Library]/[Mode]/[Mode :] option is changed.	
		~
Ŀ	$\overline{\mathcal{G}}$ Generate Upgrade.txt as a summary file in the project directory	
	OK	

(8) ワークスペースが開かれると、[Select Emulator mode]ダイアログボックスが開きます。[Device]にデバッグ対象 マイコン名称を指定し、[Mode]を[E200F Emulator]を選択して[OK]ボタンを押してください。

Select Em	ulator mode	×
<u>D</u> evice	R5E70865R	
Mode	○E200F Emulator ○ Writing Elash memory	
4	OK Cancel	

(9) [Function select]ダイアログボックスが開きます。デフォルトのまま[OK]ボタンを押してください。

F	unction select	X
	-Function setting	RealProfile Area1 (nest mode)
	<u>B</u> us board mode setting:	Trace/break 6ch (Trace 262144 (
	Option board mode setting:	RealProfile Area2 (function mode) 💌
	-Monitor setting ☞ ⊻irtual space	C Physical space
		OK Cancel

本書の例では、"R5E70865R"を選択しています。



(10)[Select Emulator]ダイアログボックスが開きます。デフォルトのまま[OK]ボタンを押してください。

Select Emulation	X
Select Emulation	
Device R5E70865R	
✓ Change Emulation in start up	
Operating mode	
Select Mode mode 3 (singlechip)	
C User system Mode	
Clock	
Emulator Internal Clock 10.0 MHz	
C User system clock	
OK キャンセル 道用(A)	

(11)[ID Code]ダイアログボックスが開きます。デフォルトの[E200F]のまま[OK]ボタンを押してください。





(12) E200F エミュレータ接続中は下図の[接続]ダイアログボックスが表示されます。



(13) E200F エミュレータの接続が完了して High-performance Embedded Workshop の画面が操作可能になります。



E200F エミュレータの接続が成功すると、アウトプットウィンドウの[Debug]タブには "Connected" が表示されます。



4.3 イベントポイントによるパフォーマンス測定

この章では、実行開始からブレークまでの測定方法と、ある区間を指定した測定方法を紹介します。

 プログラムをダウンロードするためワークスペースのロードモジュールファイル(拡張子.abs)をダブルクリックして ください。



(2) 続いてワークスペース内のソースファイル "tutorial.cpp" をダブルクリックしてソースウィンドウに表示させてくだ さい。





(3) [表示]メニューの[パフォーマンス]の中の[パフォーマンス解析]を選択してください。

🖗 Tutorial – High-p	erformance Embedde	d Workshop -	[tutor	ial.cpp	1					
	表示(い) プロジェクト(P)	ビルド( <u>B</u> ) デバック	)(D)	基本設定	È(U) y	ソール	Ē	テスト( <u>S</u> ) ウィ	ンドウ₩	ヘルプ(円)
🗅 😂 🖬 🖪 🤮	差分(E)			•	¶ #	2	\$	) 🖽 🖽 👗	debug	_e200f_system
10 10 8	MRP マップ(P)		Ð.	(+) (}	(ን 💿	J I <sub>P</sub>	PC	81 💭	1/0	🕶 📖 🖬
	🖂 コマンドライン(1)	Ctrl+L								
⊡⊸r⊒ Tutorial	🍇 TCLツールキット(Ϗ)	Ctrl+Shift+K			-		- (			
🗄 🔄 C source	ロークスペース(1)	Alt+K	Event	Aud	Bus	<u> </u>	S/	ソース		
≚ dbsc ≚intpr	🖪 アウトプット(山)	Alt+O						/********** /* /*	******	
·····································	↓ ◎□ 逆アセンブリ(D)	Ctrl+D						/* FILE /* DATE /* DESCRI	PTION	Wed, May 2
¥ vect	t CPU( <u>C</u> )	,	-					/* CPU TY	PE :	Other
⊡ 🔄 C++ sou		,						/* /*		
E-G Downloa	( グラフィック( <u>G</u> )	)						/*******	******	********
⊡[↓] Tuto ⊡ <mark>⊡</mark> Depende	パフォーマンス( <u>P</u> )		Eパ	7#-72	/ス解析	(P)	<b>b</b>	Ctrl+Shift+P		
- 🗐 sbrk. - 🗐 sort	   コード©)	)	7	บวราใม	,( <u>F</u> )			Ctrl+Shift+F	s>	nit::init
Stack	kseth 17 h 18		🗐 U7	アルタイム	,プロファ·	1N (I	ચ	Ctrl+Shift+Q	plu	IS
			_						_	
<u>.</u>	🔁 🔤 🕹 tutorial.c	pp								

(4) [パフォーマンス解析方式の選択]ダイアログボックスが表示されますので、"Onchip Performance"を選択して [OK]を押してください。

パフォーマンス解析方式の違	麗択	? 🔀
パフォーマンス解析方式( <u>P</u> ):	Onchip Performance	→          →          ☆



(5) パフォーマンス解析ウィンドウが開きます。



(6) 次に、[表示]メニューの[コード]の中の[イベントポイント]を選択してください。

🖗 Tutorial – High-p	erforma	ance Embedde	d Workshop -	[tutor	ial.cpp	]							
→ ファイル(E) 編集(E)	表示♡	プロジェクト( <u>P</u> )	ビルド( <u>B</u> ) デバッ	<u>ש) פ</u>	基本設定	定(U)	ツール	Ð	テスト(	<u>s</u> ) (	ウィンドウ\₩	ヘルプ(	<u>H</u> )
🛛 🗅 🖨 🖉 🖉	差分	}( <u>F</u> )			•	nt &	2		3 🗮	ta a	🔏 🛛 debu	g_e200f_s	ystem
16 10 8	MAP 797	1( <u>P</u> )		Ð.	(+) ()+	{}•	II) I <sub>PO</sub>	PC	R	1	1/0	FF   IIII	3
	ידב 🖂	ッドライン①	Ctrl+L						111				
E Tutorial	🗞 TOL	ツールキット( <u>K</u> )	Ctrl+Shift+K										
⊡ — 🔄 C source	<b>B</b> 9-	ウスペース( <u>K</u> )	Alt+K	Event	Aud	Bus.		S/	- ソー2 【****	र ****	******	******	*****
≝intpr	ליק 🔼	トプット(山)	Alt+O						/* /* 1	-			
≚]rese ≚]sbrk.	t 6月 逆ア	センブリ( <u>D</u> )	Ctrl+D						/* /*	DATE	RIPTION	:Wed, M :Main P	ay 22 rogra
∰ vect	t CPL	)(C)	I						/* 1 /*	CPU 1	TYPE	:Other	
sort	<u>ر</u> و ک	ボル( <u>S</u> )	I	·					/* /*				
E Downloa	( <i>75</i> :	7ィック( <u>G</u> )	I	•					/***	****	******	*****	****
⊡ <mark>↓</mark> Tuto	パフ;	ォーマンス( <u>P</u> )	I	•					ti ca	.f	colueol		
≣ sbrk. ≣ sort	-1	۴ <u>0</u> )		R I	レース( <u>T</u>	)			Ctrl+	T	e <ios></ios>	us Init::i	nit c
stac	ksct.h	17	<	<b>E</b> 0 1	ベントポイ	イント値	2		Ctrl+	E	Duspl	us	
	•				ードカバ	レジ( <u>C</u> ).		Ctrl+	Shift+	н			
- 🔄 . 🗐 . 🔍 . 🗌	<b>7</b>	🧈 tutorial.c	pp	<b>開</b> ス	タックトレ	,,. <u>(K</u> )			Ctrl+	К			
	1			_									
Channel Cond:	ition	Result											
Ch1 DISA	BLE	00000000											
Ch2 DISAN	BLE	00000000											
Ch3 DISAN	BLE	00000000											
LICh4 DISAI	BLE	0000000											



(7) イベントポイントウィンドウが表示されます。[Onchip Event]タブをクリックして任意の条件を右クリックして [Combination action(Sequential or PtoP)]を選択してください。

🖗 Tutorial – High-performan	ice Embedded Worksho	p - 1772	トポイントコ			
🧼 ファイル(E) 編集(E) 表示(⊻)	プロジェクト(P) ビルド(B) 5	デバッグ( <u>D</u> )	基本設定(U)	ッール① テスト⑤ ウィンドウW) へ	ルプ(田)	_ 8 ×
🗋 🗅 🚅 🖬 🎒 🎒 🕹 🕹 🖿	💼   😣 🛛 🗛 📃		- M 4	🎖 🐕 🛛 🏶 🖽 👗 🛛 debug_e20	00f_system 💌 💿	sessione200f_system 📃 💌
🕙 👿 <u>16</u> 10 8 2 🛒 !	👎 🔤 🕂 🔤 📑 🗄	l et el	40 <del>1</del> 6 <del>(1</del> 7	🐵 I <sub>pg</sub> 🎠 🛛 🖪 💭 🐺	📖 式 💐 🛛	<u>a</u> 20
				199	[33	
E Tutorial	a 🥒 🗙 🖃					
🖃 🖓 Tutorial	Туре	State	Conditi	on	Action	
dbsct.c	Ch1(IA OA DT CT)	Disable	None		Break	
lintpre.c	Ch2(IA_OA_DT)	Disable	None	追加( <u>A</u> )	1	
≝ sbrk.c	Ch3(IA) Ch4(IA)	Disable Disable	None None	編集(1)		
vecttbl.c	Ch5(IA)	Disable	None	有効(N)		
⊡	Ch6(IA)	Disable	None	無効( <u>S</u> )		
≝ sortcpp ≝ tutorial.cpp	Ch7 (IA)	Disable	None	削除( <u>D</u> )		
🖃 🔄 Download modu	Ch8 (IA)	Disable	None	すべてを削除( <u>E</u> )		
Tutorial.abs	chj(IA)	Disable	None	ソースを表示( <u>G</u> )		
Dependencies						
sort.h				ファイルを研じる(E) すべてのファイルを閉じる(E)		
stackset.h				9110077177219102017		
wecth	Breakpoint 🔥 On	chip Event	AUD Lient	Combination action(Sequential or Pt	toP)	
	🧼 tutorial.cpp 🗔 イベン	小术イ		✔ ツールバー表示		
				ツールバーのカスタマイズ		
× • × × •				ドッキングビュー		
Channel Condition	Result			非表示		
Ch1 DISABLE	0000000					
Ch2 DISABLE	00000000					
Ch3 DISABLE	0000000					
Ch4 DISABLE	0000000					

(8) [Combination action(Sequential or PtoP)]ダイアログボックスが表示されます。[Ch 1,2,3]に"Don't care"を選択して [OK]を押してください。



・Don't care → シーケンシャル条件および、パフォーマンスの測定開始/終了を設定



(9) パフォーマンス解析ウィンドウ上で右クリックして[設定]を選択してください。

🖗 Tutorial – High-performance Embed	ided Workshop - [イベントポイント]	
🧼 ファイル(E) 編集(E) 表示(V) プロジェクト(E	P) ビルド(B) デバッグ(D) 基本設定(U) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)	_ 8 ×
D 😅 6 🕼 🥔 🖇 🛍 🔂	🚰 🔽 🛃 🖓 🏶 🛗 📥 👗 🛛 debug_e200f_system	sessione200f_system
🕂 地 🕅 🚹 10 8 2 🛒 👫 👪 🕇	· 🔤 🛛 If Il If Il P) P P @ 1, 🔭 🖉 🗷 🖉	
Tutorial	X E State Condition Action OA DT CT) Disable None Break	
- 当 intprg.c Ch2 (IA - 달 resetprg.c Ch3 (IA) - 달 sbrk.c Ch4 (IA) - 달 vectblc Ch5 (IA) 	OA_DT) Disable None Break Disable None Break Disable None Break Disable None Break Disable None Break	
Ch9 (IA)     Ch7 (IA)     Ch8 (IA)     Ch7 (IA)     Ch8 (IA)     Ch9 (IA)     Ch9 (IA)     Ch10 (IA)     Ch10 (IA)	数定S) りたから 全てリセット(E) Break Break	
	・ データクリア(©) 全てのデータをクリア(L) ファイルに(保存	
	J     Other Event     λ     BUS Event       al(     μ     μ     μ	
Channel Condition Result	✓ ドッキングビュー 非表示	
Ch1         DISABLE         0000000           Ch2         DISABLE         0000000           Ch3         DISABLE         0000000           Ch4         DISABLE         0000000	ロロション 右クリック	

(10)[Performance Analysis]ダイアログボックスが表示されるので、[Channnel1]に"Elapsed time"を設定してください。

Performance Ar	alysis ? 🗙
Condition	1
Channel <u>1</u>	Disabled
Channel <u>2</u> 🧲	Elepted time
Channel <u>3</u>	Branch instruction counts Number of execution instructions
Channel <u>4</u>	Interrupt counts URAM area access counts
	URAM area instruction access counts URAM area data access counts

パフォーマンスの測定項目の内容は下記のとおりです。

・Disabled(オプション名なし) → 設定を解除[DISABLE]

・Elapsed time(AC) → 実行サイクル数(Iφ)を測定項目に設定

- ・Number of execution states(VS) → 実行ステート数を測定項目に設定
- ・Branch instruction counts(BT) → 分岐命令回数
- •Number of execution instructions(I)  $\rightarrow$  実行命令数
- ・Exception/interrupt counts(EA) → 例外・割り込み回数
- ・Interrupt counts(INT) → 割り込み回数
- ・URAM area access counts(UN) → URAM エリア命令・データアクセス回数
- ・URAM area instruction access counts(UIN) → URAM エリア命令アクセス回数
- ・URAM area data access counts(UDN) → URAM エリアデータアクセス回数
- (11)同様に Channel2~Channel4 までを下図のように設定して[OK]を押してください。



Performance A	nalysis	? 🔀
Condition		
Channel <u>1</u>	Elapsed time	•
Channel <u>2</u>	Number of execution states	•
Channel <u>3</u>	Branch instruction counts	•
Channel <u>4</u>	Number of execution instructions	
·	ОК	キャンセル

(12)パフォーマンス解析ウィンドウに設定した内容が表示されます。





(13) ソースファイル "tutorial.cpp"の関数 main の先頭行(28 行目)をダブルクリックして、Onchip Event ブレークを設定 してください。



(14) Onchip Event ブレーク設定後、[デバッグ]メニューの[リセット後実行]を選択してプログラムを実行してください。





(15) Onchip Event ブレーク設定した行でプログラムが停止し、パフォーマンス解析ウィンドウにパフォーマンス測定 結果が表示されます。また、アウトプットウィンドウには"EVENT CONDITION 1for L bus"のコメントが出力されます。



リセットルーチンから関数 main までのパフォーマンス測定結果を表示します。

#### 【測定結果】

・AC(実行サイクル数(Iφ)) → 31830(00007C56H)

経過時間 = 実行サイクル( $I\phi$ ) × CPU クロック周期( $I\phi$ )

= 3183 $\mu$  s

(CPU クロック周波数の設定については 4.2 章(10)を参照)

·VS(実行ステート数) → 1155(00000483H)

•BT(分岐命令回数) → 560(0000230H)

·I(実行命令数) → 1146(0000047AH)



(16) ここでいったん[Onchip Event]の条件 BC1 を削除します。



条件 BC2 が設定されている場合は同様に削除してください。

(17) ソースファイル "tutorial.cpp"の 45 行目の次に 48 行目の順番でダブルクリックしてください。次に[イベントポイント]タブをクリックしてください。





(18) イベントポイントウィンドウの任意の条件を右クリックしてポップアップメニューの[Combination action(Sequential or PtoP)]を選択してください。

🛞 Tutorial – High-performance Embedded Workshop	- [イベントポイント]	
◇ ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デ	バッグ(D) 基本設定(U) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)	_ 8 ×
	💌 🐴 🙀 😵 🕮 🛗 👗 🛛 debug_e200f_system	💌 sessione200f_system 💌
🕙 💯 16 10 8 2 🛒 🚝 🚾 🚧 📰 📑 🗉	L 🗈 🔁 🕂 🕂 🖤 💷 🛼 📗 📰 💭 🐺 📖 😭	X 🔊 🕅 🕅
		1
⊡@ Tutorial □⊜ C source file	State Condition	Action
dbsctc Ch1(IA_OA_DT_CT) E	nable Address=00001056(tutorial.cpp/45) pc Break	k Break
발 intpre.c Ch2 (IA_OA_DT) E	nable Address=00001062(tutorial.cpp/48) pc Break	k Break
Sbrk.c Ch3 (IA) D	isable None isable None	Break
vecttblc Ch5(IA) D	isable None 追加(A)	Break
Sort.cpp Ch6 (IA) D	isable None 編集(T)	Break
tutorial.cpp Ch7 (IA) D	isable None 有効(N)	Break Break
Download modu Ch9 (IA) D	isable None 無幼(S)	Break
Dependencies Ch10(IA) D	isable None  自防(型)  オペアオ客(IRA/C)	Break
	977C 2810PR/27	
stackscth	ソースを表示(G)	
vect.h	hip Event ( AUD ファイルを閉じる(C)	
	すべてのファイルを閉じる(E)	
	Combination action(Sequential or PtoP)	
× • • • • • • •	✓ ツールバー表示	
Channel Condition Result	ツールバーのカスタマイズ	
Ch1 AC 00007c56	ドッキングビュー	
ch2 vs 00000483	非表示	
Ch3 BT 00000230		
UI I OOOOOAA		

(19) [Combination action(Sequential or PtoP)]ダイアログボックスが表示されるので、[Ch 1,2,3]に"Ch 1 to Ch 2 PA" を選択して[OK]を押してください。



- ・Ch 1 to Ch 2 PA → Event Condition 1 条件(開始条件)成立から Event Condition 2 条件(終了条件)成立までの期間をパフォーマンス測定期間に設定
- ・Ch 2 to Ch 1 PA → Event Condition 2 条件(開始条件)成立から Event Condition 1 条件(終了条件)成立までの期間をパフォーマンス測定期間に設定



(20)イベントポイントウィンドウに選択した内容が表示されます。

🛞 Tutorial - High-performa	ance Embedded Workshop - M	ペントポイントコ	
	) プロジェクト(P) ビルド(B) デバッグ(D	) 基本設定(U) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)	_ 8 ×
D 🛩 🛛 🗗 🧉 🕹 🕹 🖿	8 8 8	💌 🐴 🖓 🖓 🏙 🛗 👗 🛛 debug_e200f_system	▼ sessione200f_system ▼
🖑 👿 <u>16</u> 10 8 2 🎐		1 7 7 7 9 0 🐨 👾 🔳 🗷 💭 👭 📖 😭	🥺 🔊 🖾 🛤
⊡ - 🚱 Tutorial	a 🖉 🗙 📼		
⊡ — 🔄 C source file	Type State	Condition	Action
dbsct.c	Ch1(IA_OA_DT_CT) Enable	e Address=00001056(tutorial.cpp/45) pc Break	Ch 1 to Ch 2 PA
⊥ intpre.c	Ch2(IA OA DT) Enable	Address=00001062(tutorial.cpp/48) pc Break	Ch 1 to Ch 2 PA
interesting.c	Ch3(IA) Disab.	le None	Break
wecttbl.c	Ch5(TA) Disab.	le None	Break
⊟ G++ source file	Ch6(IA) Disab.	Le None	Break
tutovialoop	Ch7(IA) Disab.	le None	Break
E Download modu	Ch8(IA) Disab.	le None	Break
↓ Tutorial.abs	Ch9(IA) Disab.	le None	Break
🗄 🔄 Dependencies	Ch10(IA) Disab.	le None	Break
≣ sbrk.h			
≣ sort.h			
) vecth	<		>
	Breakpoint Onchip Eve	ent & AUD Event & Other Event & BUS Event /	
	🥑 tutorial.cpp 🔄 イベントポイ		
× •, ×, ×, • • •			
Channel Condition	Result		
Ch1 AC	00007c56		
ch2 VS	00000483		
Ch3 BT	00000230		
Ch4 I	0000047A		

(21) [tutorial.cpp]タブをクリックし、52 行目で右クリックして[BC10]の[Add]を選択してください。





(22)パフォーマンス解析ウィンドウのパフォーマンス測定結果をクリアした後、プログラムを実行してください。





(23)52 行目のブレーク条件で停止し、パフォーマンス解析ウィンドウにパフォーマンス測定結果が表示されます。また、アウトプットウィンドウに"EVENT CONDITION 10"のコメントが出力されます。



この時の測定結果は、45 行目から 48 行目までになります。ここでは、2つの関数 sort と change を連続して呼び出した合計の測定結果となります。



4.4 データ条件(イベントポイント)によるパフォーマンス測定

この章では、データを条件とした設定例を紹介します。

(1) 変数(RAM の FFFFA000 番地)にデータ(ここでは"H'00001234")が書かれたときを条件とした設定例を紹介します。イベントポイントウィンドウの[Ch1]欄をダブルクリックすると、[Event condition 1]ダイアログボックスが表示されるので、[Address]タブをクリックして[Address]を"H'FFFFA000"に設定してください。



[Data]タブをクリックして[Value]を"H'00001234"に設定してください。

Event condition 1	? 🗙
Address Data Bus State Count Action	
Data	-
☐ <u>D</u> on't Care	
Value H'00001234	
C <u>B</u> yte C <u>W</u> ord I <b>●</b> Long	
⊙ <u>N</u> on user mask ⊂ <u>U</u> ser mask	
Mask	
	( <u>A</u> )

[Bus State]タブをクリックして[Bus type]を"L-Bus(CPU)"、[Read/Write]を"Write"に設定して[OK]を押してください。

Event condition 1	? 🗙
Address Data Bus State Count Action	
Bus state Bus type	
☐ <u>O</u> PU(I-Bus) ☐ <u>D</u> MA(I-Bus)	
DIC(I-Bus)	
Read/Write C Read/Write C Bead O™rite	
	i用( <u>A</u> )



(2)次に、条件(2)として変数(RAM の FFFFA100 番地)にデータ(ここでは"H' 000055AA")が書かれたときを条件とした設定例を紹介します。イベントポイントウィンドウの[Ch2]欄をダブルクリックすると、[Event condition 2]ダイアログボックスが表示されるので、[Address]タブをクリックして[Address]を"H' FFFFA100"に設定してください。

Event condition 2	? 🔀
Address Data   Bus State   Action   Address Don't Care • Address	
Only program fetched address     Only program fetched address after	
Address HFFFFA100	
ি Non <u>u</u> ser mask ⊂ U <u>s</u> er mask Mask	
	適用( <u>A</u> )

[Data]タブをクリックして[Value]を"H'000055AA"に設定してください。

Event condition 2	? 🗙
Address Data Bus State Action	
Data	
C <u>B</u> yte C <u>W</u> ord ⊙ <u>L</u> ong	
<u>M</u> ask	
	i用( <u>A</u> )

[Bus State]タブをクリックして[Bus type]を"L-Bus(CPU)"、[Read/Write]を"Write"に設定して[OK]を押してください。

Event condition	2 ? 🛛
Address Data	Bus State Action
– Bus state <u>B</u> us typ <mark>e</mark>	
	ead/Write ead frite
	OK キャンセル 適用(ム)



(3) パフォーマンス解析ウィンドウに設定した内容が表示されます。

Type	State	Condition Action			
Ch1(IA_OA_DT_CT)	Enable	Address=FFFFFA000 data H'1234 long access L-bus Ch 2 to Ch 1 PA			
Ch2(IA_OA_DT)	Enable	Address=FFFFFA100 data H'55aa long access L-bus Ch 2 to Ch 1 PA			
Ch3 (IA)	Disable	None Break			
Ch4 (IA)	Disable	None Break			
Ch5(IA)	Disable	None Break			
Ch6 (IA)	Disable	None Break			
Ch7(IA)	Disable	None Break			
Ch8 (IA)	Disable	None Break			
Ch9(IA)	Disable	None Break			
Ch10(IA)	Disable	None Break			
ζ					
Breakpoint Onchip Event (AUD Event ) Other Event BUS Event /					
✓ tutorial.cpp ☐ イベントボイ					

上図のようなデータ条件の測定も可能です。

### 4.5 実行命令数を使ったパフォーマンス測定の応用例

この章では、ある1命令に対しての実行回数を測定する方法を紹介します。

(1) ソースファイル"tutorial.cpp"内の行番号 48 と 49 の間の[Onchip Event]欄をダブルクリックして Onchip Event ブレークを設定します。



連続する1命令の前後を測定条件としてI情報(実行命令数)を見ることにより、該当部分の通過回数を知ることが可能です。例えば、割り込み処理ルーチン内に設定することにより、割り込み発生回数を測定することができます。



(2) イベントポイントウィンドウに設定した内容が表示されます。

Туре	State	Condition	Action
Ch1 (IA_OA_DT_CT) Ch2 (IA_OA_DT) Ch3 (IA) Ch4 (IA) Ch5 (IA) Ch6 (IA) Ch6 (IA) Ch7 (IA) Ch8 (IA) Ch9 (IA) Ch10 (IA)	Enable Enable Disable Disable Disable Disable Disable Disable Disable	Address=00001062(tutorial.cpp/48) pc Break Address=00001064(tutorial.cpp/48) pc Break None None None None None None None None	Ch 1 to Ch 2 PA Ch 1 to Ch 2 PA Break Break Break Break Break Break Break Break
Breakpoint      Onchip Event      AUD Event      Other Event      BUS Event			
□ イベントボイ			

アドレス H' 00001062 番地から H' 00001064 番地を測定条件とします。

(3) プログラムを実行することにより、パフォーマンス解析ウィンドウにパフォーマンス測定結果が表示されます。ある1命令のIRQ発生回数を測定することで実行回数を測定することができます。

×	× + × × × • • • • • •			
	Channel	Condition	Result	
	Ch1	AC	0000F394	
	Ch2	VS	000079CA	
1	Ch3	вт	00000000	
	Ch4	I	000079CA	

この場合、実行命令数は H' 79CA(D' 31178)となり、アドレス H' 00001062 番地を 31178 回通過したことになります。

以上で、パフォーマンス測定方法の説明を終わります。

# RENESAS

## 5. よくある質問

#### 5.1 パフォーマンス測定を行う上で注意する点はありますか?

①オンチップパフォーマンス測定機能とプロファイル機能は、同時に使用できません。同時に使用しようとした場合、「Can not use this function」エラーメッセージダイアログボックスが表示されます。

②ユーザプログラム実行中に Onchip イベントのチャネル 1、チャネル 2 を用いたパフォーマンス測定機能 (Combination action(Sequential or PtoP)ダイアログ上の[Ch1,2,3]の[Ch 2 to Ch 1 PA]設定または[Ch 1 to Ch 2 PA]設定)を変更しないでください。変更した場合、正しくパフォーマンス測定ができません。

③パフォーマンス測定にて終了条件成立後に、開始条件が成立した場合は、パフォーマンス測定を再開します。ブレーク後の測定結果は、パフォーマンス測定期間中の測定結果の合算になります。

④パフォーマンス測定開始/終了(Ch2 to Ch1 PA および Ch1 to Ch2 PA)を使用する場合、Event Condition 1 条件の回数指定は 1 回としてください。

⑤DMA または DTC 転送と外部バスアクセス条件を含む Event Condition 条件の成立が競合した場合、この Event Condition 条件成立によるブレーク、内蔵トレースのトレース停止およびトレース取得、パフォーマンス の測定開始/終了が動作しないことがあります。

⑥SH7125/SH7124のエバチップユニット接続なしでのデバッグの場合、パフォーマンス測定機能をサポート していません。SH7125/SH7124のエバチップユニット接続ありでのデバッグの場合、パフォーマンス測定機 能をサポートしています。

# RENESAS

#### 6. 関連ドキュメント

E200F エミュレータおよび、HEW には本書で取り上げた機能以外にも便利な機能を豊富に備えています。各製品の仕様の詳細、技術情報、制限事項など有用な情報を記載していますので下記の関連ドキュメントも合わせて参照してください。

#### 【E200F エミュレータ関連ドキュメント】

- ・ SH-2A、SH-2 E200F エミュレータユーザーズマニュアル
- SH-2A SH-2 E200F エミュレータ ご使用上の注意事項
- ・ SH-2A、SH-2 E200F エミュレータユーザーズマニュアル 別冊 SH7086 ご使用時の補足説明

【High-performance Embedded Workshop 関連ドキュメント】

- ・ High-performance Embedded Workshop ユーザーズマニュアル
- ・ High-performance Embedded Workshop リリースノート

#### 【CPU 関連ドキュメント】

- ・ SH7080 グループ ハードウェアマニュアル
- ・ SH-2A、SH2A-FPU ソフトウェアマニュアル

#### 【SuperH ファミリ用 C/C++コンパイラパッケージ関連ドキュメント】

- ・ RISC engine C/C++コンパイラ、アセンブラ、最適化リンケージエディタ コンパイラパッケージューザー ズマニュアル
- SuperH RISC engine C/C++ コンパイラパッケージ アプリケーションノート

#### 本製品に関する情報は以下のルネサス・ウェブサイトをご覧下さい:

日本サイト:	http://japan.renesas.com/e200f
グロバールサイト:	http://www.renesas.com/e200f



# ホームページとサポート窓口

ルネサステクノロジホームページ http://japan.renesas.com/

お問合せ先

http://japan.renesas.com/inquiry csc@renesas.com

# 改訂記録

		改訂内容		
Rev.	発行日	ページ	ポイント	
1.00	2007.10.25	—	初版発行	



■ 本資料ご利用に際しての留意事項 ■

本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載 1. の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するもので はありません。 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に 2. 起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途 3. の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令 を遵守し、それらの定めるところにより必要な手続を行ってください。 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資 料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがありま 4. す。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認 頂きますとともに、弊社ホームページ(http://www.renesas.com)などを通じて公開される情報に常にご注意 下さい。 本資料に記載した情報は、正確を期すため慎重に制作したものですが、万一本資料の記述の誤りに起因する 5. 損害がお客様に生じた場合においても、弊社はその責任を負いません。 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例 6. などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、 お客様の責任において適用可否を判断して下さい。弊社は、適用可否に対する責任は負いません。 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海 7. 底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれ のあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設 計、製造されたものではありません (弊社が自動車用と指定する製品を自動車に使用する場合を除きます)。 これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用 途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使 8. 用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。 生命維持装置。 1) 人体に埋め込み使用するもの。 2) 治療行為(患部切り出し、薬剤投与等)を行なうもの。 3) 4) その他、直接人命に影響を与えるもの。 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびそ 9. の他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場 合の故障および事故につきましては、弊社はその責任を負いません。 10. 弊社は製品の品質及および信頼性の向上に努めておりますが、特に半導体製品はある確率で故障が発生した り、使用条件によっては誤動作したりする場合があります。弊社製品の故障または誤動作が生じた場合も人 身事故、火災事故、社会的損害などを生じさせないよう、お客様の責任において冗長設計、延焼対策設計、 誤動作防止設計などの安全設計(含むハードウエアおよびソフトウエア)およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウエアは、単独での検証は困難な ため、お客様が製造された最終の機器・システムとしての安全検証をお願い致します。 11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険 性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において 十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を 負いません 12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致し ます。 13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご 照会下さい。

© 2007. Renesas Technology Corp., All rights reserved.