付録基板を120%使いこなすために SH-2 基板用JTAGデバッガ登場!

内藤 竜治

本誌 2006 年 6 月号に付録した SH-2 基板上で動作する JTAG デバッガを開発しました(図1).

この JT AG デバッガを使えば,どなたでも SH-2 上で JT AG を使ったデバッグを体験できるようになります.なお,この JT AG デバッガは本誌 2006 年 6 月号付録の SH-2 基板(SH7144F) でのみ動作します.

1. JTAG デバッガとは

JTAG を使って CPU をデバッグ

JTAGとは, ICの中のテスト回路と通信するためのシリアル 通信の標準規格です.CPLDや FPGAの書き込み用によく使わ れるので,馴染みの深い方も多いでしょう.

JTAGデバッグというのは, JTAGを使ってターゲット CPU の内部のレジスタを見たり,メモリの内容を書き換えたり, CPU をステップ実行させるといったデバッグ方法です.また, そのようなデバッグを行うためのソフトウェアのことをJTAG デバッガといいます.パソコン上のデバッガ・ソフトウェアと, パソコンとターゲット・ボードをつなぐハードウェアのことを あわせてJTAG デバッガと呼ぶこともあります.

JTAG デバッガは,ターゲット・ボードの ROM にモニタを 常駐させるデバッガとは違い,ターゲット上の ROM や RAM と いったユーザ用のリソースを占有しません.JTAG デバッガを 使った場合,CPU のメモリ空間は最終的な製品と同じように使 うことができるので,実機に近い状態でのデバッグが可能です.

なお,ルネサス テクノロジ製の CPU では,JTAG ではなく H-UDI(Hitachi User Debug Interface)という名称で呼ばれるこ とも多いのですが,本記事ではJTAG という名称を使います.

本デバッガの機能

本デバッガを使うと, SH-2 に対して表1に掲げるようなデ

					<u> </u>	間 GPI	U Registers		<u> </u>
Address	Code	Asm		Comments	<u>^</u>	RO	FFFF401C	PC	0000081C
00000810	644C	EXTU.B	R4,R4			R1	00000010	PR	0000081C
00000812	440E	LDC	R4,SR			R2	FFFF8A54	GBR	FFFFFFE
00000814	0009	NOP				R3	FFFF8A52	VBR	00000000
00000816	D506	MOV.L	@(6,PC),R5			R4	000000F0	MACH	82410241
0000818	450B	JSR	@R5			R5	00001000	MACL	80FF0000
0000081 A	0009	NOP				R6	00000000	SB	000000E0
000081C	001 B	SLEEP				87	EEEE4330		
0000081 E	000B	RTS				DO			
0000820	0009	NOP				100	CPU0.	レシ.	スダの
JUUUU822	0008	RIS				R9	値が表	示され	13 📋
00000824	0009	NOP	旧たっぱっぱ=		****	R10		1	
0000826	0000	111	現住のノロクラ	な・カウノタの	削後の	R11	00000000		
00000828	0000	777	逆アセンブル・	リストが表示さ	れる	R12	0007A11F		
JUUUU82A	0010	777 L				R13	0001 E847		
JUUUU82C	1000	1111	00.0 (0.00)			R14	FFFFE274		
JUUUU82E	1028	MOV.L	H2,@93,HU7			R15	FFFFFFF0		
00000830	1000	TTT MON (I	00.0/0.00)			(02) (5)	and t	-	
00000832	1000	MOV.L	R0,@10,R07			FEEEE	EE0 0~0000000	0	티비스
00000834	FFFF	000				FFFFF	FF4_0x0000000	Ď	
00000830	FFFF	000			×	FFFFF	FF8 FFC フタ	<i>ω π</i>	山良が
000380						10		+ + + > >	T-3/J
Memory du	mo start	-			and the second		衣小	2112	· .
0800 D6 0	9 E1 10 36	18 46 2E D4	08 44 0B 00 09 E4 F0 3.	· 6 5 6 0				000000	等 2h
0810 64 4	C 44 0E 00	09 D5 06 45 09 00 00 00	08 00 03 00 18 00 08 dL 00 00 10 00 00 10 28	♪ メモリのダ	ンプ結果など	Ľ	0 00	000000	
1870 101 0						_			

図1 本デバッガのメイン画面

付録基板を120%使いこなすために SH-2 基板用JTAGデバッガ登場!

表1 本デバッガの機能

CPU の停止(ブレーク)と再開(リスタート)
メモリの読み書き
レジスタの読み書き
シングル・ステップ実行
ハードウェア・ブレークポイント設定
SH-2 CPU のリセット
実行しているプログラムの逆アセンブル
任意のアドレスへのジャンプ
スタックの表示
メモリ・ダンプ
GDB との接続(GDB Stub 機能)
ソフトウェア・ブレークポイントの設定
(GDB から操作する場合に限り設定可能)
GCC で作成した ELF ファイルのダウンロード
そのほか,任意のバイナリ・ファイルのダウンロード

GND 電源(3.3~6V)入力 12 38) /40 GND $\bigcirc \bigcirc 13$ \bigcirc V_{ref} \bigcirc RES ---- $\bigcirc \bigcirc$ TDI $\bigcirc \bigcirc$ \frown SH7144 TMS ASEBREAK (--) TDO-O Q \bigcirc TRST тск 図2 コネクタ J3 のピン配置

バッグを行うことができます.

本デバッガのコアとなる部分では,SH-2の中にあるデバッ グ・ユニットと通信をして,CPUの汎用レジスタの読み書き, メモリの読み書き,CPUの動作の停止と再開,ブレークポイン トの設定,ステップ実行などデバッグの基礎となる機能を提供 しています.

これらのコアとなる機能をいくつか組み合わせることで,高 度な機能を実現しています.たとえば,プログラムのダウン ロードと実行は,メモリへの書き込みとプログラム・カウンタ (PC)への値のセットで実現できます.

2. 接続ケーブルと JTAG 信号の接続

準備するもの

本デバッガを使うには,Windows2000またはXPが動作する パソコンと本ソフトウェア,付録基板 CQ7144A ボードと, JTAG 接続ケーブルが必要です.これらのものがあれば,RAM にプログラムをダウンロードしてデバッグすることができます.

CPU の内蔵フラッシュ・メモリにプログラムをダウンロード してデバッグするには,これに加えて RS-232-C と FDT(Flash Development Toolkit 3.05)が必要です.

JTAG 接続ケーブルには, Xilinx 社の Parallel Cable III また は IV,もしくは Altera 社の ByteBlaster MV または II ケーブル が利用できます.これらの JTAG ケーブルは簡単に互換品を自 作することもできます.

ノート・パソコンなどでプリンタ・ポートがないという方は, USBやRS-232-Cでも接続することができるようにしています(詳細は稿末のURLを参照).

パソコンと SH-2 基板の接続

パソコンと SH-2 をつなぐためには JT AG の信号を使います. CQ7144A 基板では J3 と書かれたコネクタに JT AG の信号が集 約されています.J3 のピン配置は図2のようになっているので, 使用する JTAG ケーブルから出ている *V_{cc}(V_{ref})*, TCK, TDI, TMS, TDO, GND の6本の線を, J3の対応する場所に1本1 本つなぎます.J3の TRST と ASEBRK の信号はオープン(基 板上でプルアップされている)のままにしておきます.

JTAG デバッグを行う場合は SH-2の DBGMD の端子は"H" レベルになっている必要があります.CQ7144A はデフォルト の状態では DBGMD はプルアップされているので問題ありませ んが,何らかの改造をした方は元に戻すようにしてください.

3. キー・コードの設定

キー・コードとは

今までに JTAG デバッガを使ったことがある人ならば,フ ラッシュ・メモリにプログラムを書き込まなくてもいきなり CPU のデバッグができるだろうと思うかもしれません.ところ がこれは組み込み機器にとってのセキュリティ・ホールとなり ます.

そこで SH7144F では,内蔵フラッシュ・メモリに第三者が 作ったプログラムが格納されている場合には,フラッシュ・メ モリ上のプログラムをJTAGでデバッグできないようになって います.JTAGデバッガが起動した際に,内蔵フラッシュ・メ モリ上のある領域に書き込まれたデータ(キー・コード)と, JTAGデバッガから送られてきた値が一致しない場合には,内 蔵フラッシュ・メモリは自動的に全消去されてしまいます.こ のようにして,自分で作ったプログラムはデバッグできるが, 他人のプログラムや製品はデバッグできないというしくみが実 現されています.

表2は,SH7144Fのメモリ・マップのうち0x00000000番地 付近を抜粋したものです.このあたりは割り込みベクタなどが 記載される領域で,内蔵フラッシュ・メモリの領域です.アド レス0x00000020番地はシステム予約とされていますが,この 番地にキー・コードを書き込みます.