

付録基板を120%使いこなすために SH-2 基板用 JTAG デバツガ登場!



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 デバツガを開発しました(図 1)。

この JTAG デバツガを使えば、どなたでも SH-2 上で JTAG を使ったデバツガを体験できるようになります。なお、この JTAG デバツガは本誌 2006 年 6 月号付録の SH-2 基板(SH7144F)でのみ動作します。

1. JTAG デバツガとは

JTAG を使って CPU をデバツガ

JTAG とは、IC 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 デバツガというのは、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ツガ方法です。また、そのようなデバツガを行うためのソフトウェアのことを JTAG デバツガといいます。パソコン上のデバツガ・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 デバツガと呼ぶこともあります。

JTAG デバツガは、ターゲット・ボードの ROM にモニタを常駐させるデバツガとは違い、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 デバツガを使った場合、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、実機に近い状態でデバツガが可能です。

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。

本デバツガの機能

本デバツガを使うと、SH-2 に対して表 1 に掲げるようなデ



図 1
本デバツガのメイン画面

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

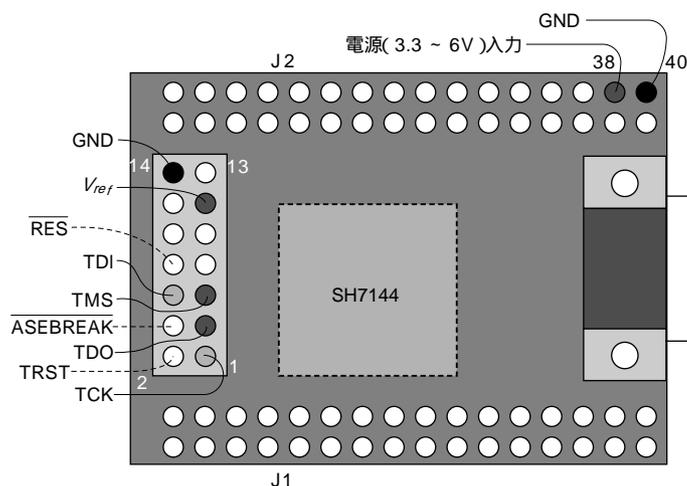


図2 コネクタ J3 のピン配置

バ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 ByteBlasterMV 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ますようにしています(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 JTAG の信号を使います。CQ7144A 基板では J3 と書かれたコネクタに JTAG 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が、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。