

第3章 多様化したCPUのJTAGデバッグ

～BDM, EJTAG, N-wire, Nexcus, …～

由良 守



JTAG機能を搭載するCPUが増えている。ただし、CPUのJTAGポートは、バウンダリ・スキャン・テストではなくオンチップ・デバッグが主目的になる。従来のICEを代替するものとして、JTAG+デバッグが期待されている。ただし、本来のJTAGポート(5本)だけではリアルタイム・トレース能力が弱い。そこで、トレース能力を強化するために、さらに信号線を増やして対処するようになった。そのため、JTAGポートといっても、デバッグ用ポートとしては多種多様な規格が乱立している。

(編集部)



1. オンチップ・デバッグの誕生の背景

■ICEの限界

マイクロプロセッサを組み込んだシステムのソフトウェア開発には、従来からICE(In-circuit Emulator)が幅広く用いられてきた。ところが半導体の微細加工技術の進歩により、マイクロプロセッサは高速・大規模化、多様化がすすみ、次のような理由から、既存のICEでは対応できなくなりつつある。

- (1)動作周波数の高速化による遅延やノイズが無視できなくなった。とくに、実機ポートと接続するケーブルのインピーダンスを整合させたり、信号遅延を最小にすることが困難になった(実現できても大きなコストがかかる)。
- (2)表面実装の採用が増え、パッケージもDIPやPGAだけでなく、フラット・パッケージやCSP, BGAなどと多様化し、CPUターゲットのソケットの製作が容易ではなくなった。
- (3)プロセッサが高速化し、ブレイクポイントやトリガ・ポイントの一致検出や大小比較などの回路を外部回路で構成すると間に合わなくなった。
- (4)バス性能ネックを解消するためにキャッシュ・メモリを内蔵するプロセッサが増えてきたが、その場合、キャッシュのヒットしている間のプログラムの履歴がとれない。
- (5)SOC時代になり外部にアドレス・バスやデータ・バス制御信号といったICEを構成するために必要な信号が端子に出なくなってきた。また出力されている場合も、ピン配置がパッ

ケージにより異なり、ICEのプロープを作り直す必要が出てきた。

■半導体メーカー側の問題点

上記はICEの開発側の問題であるが、半導体メーカー側の問題も無視できなくなった。

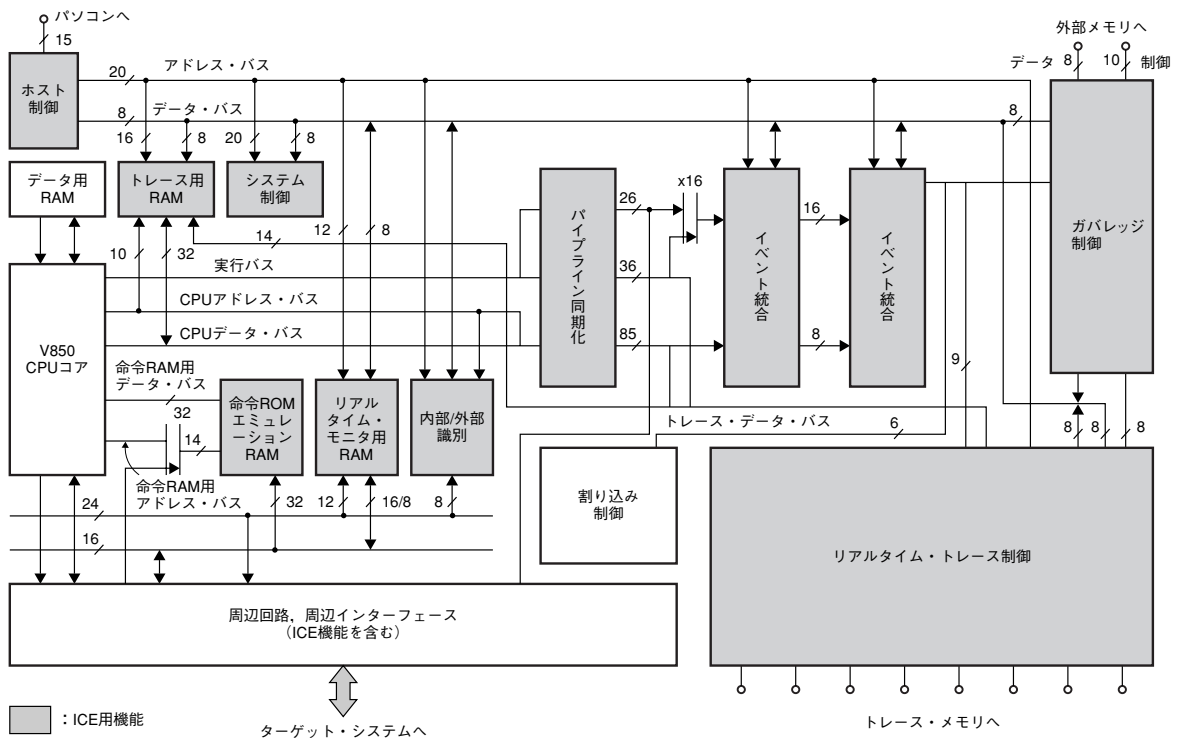
- (1)従来、プロセッサのデバッグ用に一般ユーザー向けではないデバッグ専用チップ(エバチップという。図1)を開発し、さらにICE開発用に量産チップとは別に評価用チップも開発してきたが、費用対投資効果が得られない(図2)。
- (2)SOC時代になり、マイクロプロセッサもセミカスタム/カスタムLSIが多くなり、多品種少量時代になった。回路規模もきわめて大きく複雑となり、コストが一層かかるうえ、製品のライフサイクルは概して短くなった。開発のリスクも高まり、採算ベースの観点から、ICEなど開発環境への投資が行いにくくなってきた(家庭用のゲーム機のように、大量生産が見込まれると話は別だが)。
- (3)CPUの周辺機能を取り込むいわゆる1チップ化がすすむにつれ、ICEもいっそう複雑になり、その開発はコストと時間がかかるようになった。CPUのライフサイクルが短くなる傾向もあり、従来型のICEで市場の要求に応えることがむずかしくなった。

■デバッグに必要な機能とは

マイクロプロセッサの応用機器(組み込み機器)のデバッグ・ツールの原点にかえてみると、従来、ICEなどのリアルタイムのデバッグ機能は次の三つに大別できる。

(1)実行制御(Run制御)

デバッグのもっとも基本的な機能で、実機上でアプリケーション・プログラムを実行開始したり、強制ブレイクやブレイクポイントにより一時停止させたりする機能のことである。自動車でたとえると、その走る、曲がる、止まるという基本操作に対応する。これに付随してプロセッサの内部の資源(レジスタなど)の読み出し/書き込みや、メモリの読み出し/書き込み、ブレイクポイントの設定/解除などを行う。



【図1】エバチップのブロック図（V850用の例）

(2)リアルタイム・トレース

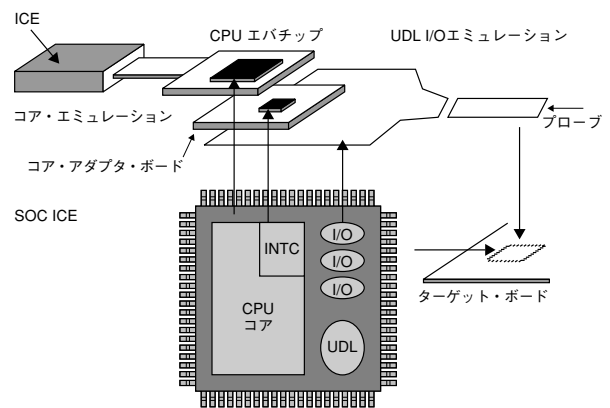
実行制御と異なり、アプリケーション・プログラムを実行させ、その実行の履歴やデータのアクセス、割り込みの状況などをトレース・メモリに記憶させ、不具合の解析を行う機能である。トレース・メモリへの書き込みの際には、書き込みを開始するためのトリガ条件やある指定された条件時のみ書き込みを行うクオリファイ・トレースなどがある。通常はアドレス・バス、データ・バス、バスのステータスや割り込み、DMAの端子などがトレースされる。書き込みのサイクルはバス・サイクルの単位、クロック単位など1kサイクルから64kサイクル（ICEのシステムにより異なる）のトレースができる。

(3)計測・データ解析

これはアプリケーション・プログラムのデバッグには直接関係がない機能で実行時間計測、実行の頻度の計測やメモリのアクセスがあったかなどのデータを収集する機能である。実行時間の計測では、2点間の時間測定や指定された処理の実行時間の最大、最小、平均値などを測定できる。またアドレス・バスやデータ・バスなどのトレース・データとともにタイマの値を同一のトレース・フレームの中に書き込むタイム・スタンプがある。

■オンチップ・デバッグとは

これらの問題点を解決するためにマイクロプロセッサ自身にデバッグ機能を埋め込むというオンチップ・デバッグ（On-chip Debug：OCDと略される）という手法が考えられた。オンチップ・デバッグは、マイクロプロセッサにデバッグ機



【図2】エバチップを用いたシステム・オン・チップのICE

プロセッサ・コア部は標準品にCPUコアのICEを用い、I/Oやユーザの回路（UDL）は専用のボードを用意する（通常はツール・メーカかユーザが用意する）。

能を組み込んだものの総称であるが、現状ではこのOCDはマイクロプロセッサの供給メーカにより数種の方式がある。たとえば、BDM、JTAG、EJTAGやN-wire方式である。

いずれにしてもオンチップ・デバッグ機能はおもに三つのレベルに分けられる。

(1)デバッグの基本機能

ソフトウェア・デバッグを行うための基本的な機能で、従来のデバッグ・モニタとほぼ等価なレベルである。プロセッサの全リソースに対してすべてアクセスできる（CPUレジスタ、周辺特殊レジスタ）。

(2)リアルタイム・トレース機能

(3)その他の付加的機能（トリガの出力）