

Tecniche di trace avanzate per l'analisi di applicazioni su Cortex-M

Parte I - Descrizione generale

I Cortex-M sono ricchi di funzionalità di debug e trace molto avanzate ma poco note. Lauterbach TRACE32 Combiprobe è il sistema a basso costo che meglio le supporta

Marco Ferrario
Lauterbach Italia

Uno degli approcci più diffusi per il test del software consiste nel generare informazioni diagnostiche sugli eventi più rilevanti che si verificano nel sistema, ed emettere tali informazioni su un canale di comunicazione esterno – tipicamente un'interfaccia seriale o Ethernet – oppure salvandole su un file di logging. Di norma le informazioni sono codificate come stringhe di testo, direttamente leggibili dal progettista e programmate usando le tipiche funzioni di gestione dello standard output o degli stream: ad esempio, in linguaggio C, le funzioni `printf()` o `fprintf()`. Si parla quindi di “debug con le `printf()`”. Questo approccio presenta diversi inconvenienti: comporta infatti una pesante strumentazione del codice sorgente, sia per la necessità di formattare in ASCII le informazioni testuali e numeriche, ad esempio necessarie a registrare il valore corrente di una variabile, sia perché richiede un canale di trasporto o un supporto di memorizzazione la cui gestione può essere molto impegnativa a livello software. Si pensi ad esempio alle tipiche velocità di trasmissione su un'interfaccia seriale RS-232, notevolmente inferiori alla velocità con cui una CPU esegue il programma e genera le informazioni diagnostiche.

Inoltre, nel caso di sistemi operativi multitasking, lo stesso canale di trasmissione o file di logging viene normalmente usato da più task in esecuzione concorrente, con proble-



matiche di sincronizzazione nell'accesso a una risorsa condivisa.

Si può dunque affermare che questa tecnica di debug, pur garantendo una visione dinamica sull'evoluzione del programma e la capacità di comprendere cosa accade all'interno del proprio sistema software, ha dei costi notevoli in

termini di progetto e implementazione, e può alterare significativamente il realtime di esecuzione, rallentando l'esecuzione del programma con possibili effetti negativi sulle prestazioni.

La soluzione

Il sovraccarico di gestione dovuto al canale di comunicazione può essere fortemente ridotto se una CPU offre una porta di trace per emettere informazioni sullo stato del sistema.

Sulla base di questa idea, molti silicon-vendors hanno realizzato soluzioni di trace dati che vengono generalmente classificate come "System Trace". Sono state realizzate soluzioni proprietarie, specifiche di un singolo fornitore, oppure basate su enti di standardizzazione fra i quali si ricorda STP (System Trace Protocol) del "Test and Debug" workgroup di MIPI Alliance.

In tabella 1 vengono mostrate le più diffuse implementazioni esistenti e il corrispondente protocollo di riferimento.

Tabella 1 - Implementazioni System Trace e protocollo di trasmissione			
Name	Vendor	Protocol	Implementato
eXtended Trace Interface	Nokia	XTI	-
Instrumentation Trace Macrocell	ARM	-	CoreSight component
System Trace Macrocell	ARM	MIPI STPv2	CoreSight component
System Trace Module	ST Ericsson	MIPI STPv1	-
PTI Output Agent	Intel	MIPI STPv1	Penwell
MIPI Trace Module	Infineon	MIPI STPv1	-
System Debug Trace Interface	Texas Instruments	XTI	OMAP3xxx AM3xxx
System Trace Module	Texas Instruments	MIPI STPv1	OMAP4xxx OMAP5xxx

Un esempio di utilizzo

Nell'articolo si prenderà in considerazione una soluzione ARM implementata nell'ambito della tecnologia CoreSight. Utilizzando come riferimento una EVB Freescale TWR-K60N512, basata su MCU Kinetis MK60DN512VMD10, saranno mostrate le possibilità offerte da un System Trace di tipo CoreSight ITM (Instrumentation Trace Macrocell). Come sistema di debug e trace sarà utilizzato un PowerTool universale Lauterbach a cui sarà collegato uno speciale debug-cable chiamato CombiProbe.



Fig. 1 - Sistema TRACE32 PowerDebug USB CombiProbe

CombiProbe

Il CombiProbe offre le tipiche soluzioni di debug possibili con un debugger Lauterbach e in più include 128-MByte di memoria trace. Può operare a una frequenza di clock di debug fino a 100MHz e fornisce una porta trace a 4 pin con data rate fino a 200Mbit/s per pin. Le prestazioni offerte dal CombiProbe sono riassunte in tabella 2. Come indicato

Tabella 2 - Protocolli System Trace supportati da TRACE32 CombiProbe	
Feature	Description
JTAG	Supports JTAG debugging like a regular JTAG debug cable. Additionally supports to debug two physically separated JTAG ports.
Serial Wire	Supports ARM's Serial Wire Debug (SWD) port and ARM's Serial Wire Viewer (SWV) output.
Infineon DAP	Supports Infineon's Device Access Port (DAP) in 2-pin configuration.
IEEE 1149.7	Supports the IEEE 1149.7 (also known as CJTAG) two-wire debug protocol.
TraceTrace	Generic support for any kind of synchronous 4-pin trace (one clock + 4 data pins) Specifically supports the "System Trace Protocol" (MIPI Alliance specification).
Trace	Specifically supports to trace ARM's Instrumentation Trace Macrocell (ITM) and Embedded Trace Macrocell (ETM), exported by a Trace Port Interface Unit (TPIU) in 4/2/1-bit continuous mode (with enabled Formatter). Specifically supports to trace ARM's ITM protocol, exported via ARM's SerialWireOutput in UART mode.
PIPE mode	Support for a PIPE mode in which trace data can be transferred on the fly to user specific applications, while the trace recording is turned on.

Tabella 3 - Debug port supportate da TRACE32 CombiProbe

Pin	JTAG Debug Port		cJTAG Debug Port		SWD Debug Port		Internal Pull-up/Down
	Type	Description	Type	Description	Type	Description	
JTAG_TMS/ SWD_DIO	I/O	JTAG Test Mode Selection	I/O	cJTAG Data	I/O	Serial Wire Data	Pull-up
JTAG_TCLK/ SWD_CLK	I	JTAG Test Clock	I	cJTAG Clock	I	Serial Wire Clock	Pull-down
JTAG_TDI	I	JTAG Test Data Input	-	-	-	-	Pull-up
JTAG_TDO/ TRACE_SWO	O	JTAG Test Data Output	O	Trace Output over a single pin	O	Trace Output over a single pin	N/C
JTAG_TRST_b	I	JTAG Reset	I	cJTAG Reset	-	-	Pull-up

in tabella, il CombiProbe è in grado di gestire diversi protocolli di System Trace. L'informazione può essere salvata nella memoria di trace fornita dal CombiProbe stesso e analizzata con i tipici comandi forniti dalla GUI TRACE32, oppure mediante apposite funzioni disponibili nelle API per il controllo remoto di TRACE32.

Per superare le limitazioni dovute alla quantità di memoria disponibile nel CombiProbe, è anche possibile attivare una modalità di funzionamento "PIPE" in cui il flusso dati viene inoltrato all'host. In questo modo la memoria del CombiProbe è usata solo come buffer. A livello di host, il software TRACE32 permette di salvare i dati di trace su file, inoltrarli a un'ulteriore pipe aperta su host o inviarli a una libreria sviluppata dall'utente (DLL / shared object) per elaborazioni più specifiche. La libreria può visualizzare i risultati in apposite finestre di TRACE32 (custom DLL) oppure effettuare ogni sorta di elaborazione proprietaria (PIPE DLL). In modalità PIPE, il software TRACE32 decodifica in tempo reale i dati formattati provenienti dal target, facendone una prima elaborazione. È anche possibile salvare semplicemente i dati di trace su hard disk senza elaborazione (modalità STREAM). Se il target implementa un ETM, è infine possibile trasferire su host e contemporaneamente analizzare i dati ETM sul flusso di programma (RTS, Real Time Streaming o Profiling).

Architettura di debug e trace ARM CoreSight

L'MCU K60N512 si basa su un core ARM Cortex-M4 e implementa un'architettura di debug di tipo ARM CoreSight. Supporta le seguenti interfacce di debug, tutte gestibili via CombiProbe:

- IEEE 1149.1 JTAG
- IEEE 1149.7 JTAG (cJTAG)
- Serial Wire Debug (SWD).

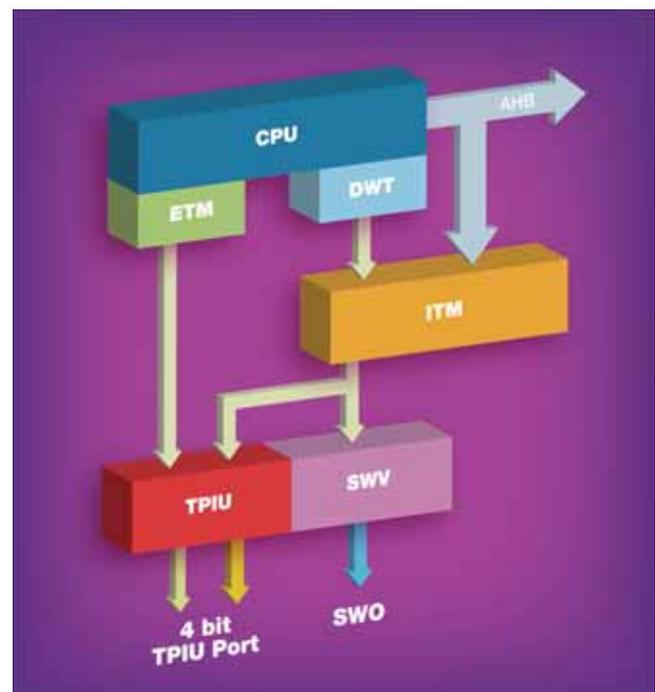


Fig. 2 - Componenti CoreSight Trace

In tabella 3 vengono riassunte le debug port supportate da TRACE32 CombiProbe. Dal punto di vista trace, i componenti CoreSight implementati sono indicati nella figura 2, che mostra anche le interconnessioni possibili fra le diverse funzioni. Di seguito una breve descrizione di ciascuno di questi componenti.

ETM (Embedded Trace Macrocell)

Genera informazioni sull'attività del core. Nel caso più generale si tratta di informazioni sul flusso di programma o di dati, ma nel caso Cortex-M è possibile emettere solo informazioni sul flusso di programma.

TPIU (Trace Port Interface Unit)

L'output di un ETM viene inviato a una porta di trace, dalla quale può essere estratto con un preprocessore di trace Lauterbach (tipo LA-7992 o LA-7993) oppure mediante CombiProbe. L'interfaccia di uscita formatta i dati d'ingresso che possono provenire da diverse sorgenti di trace (ETM e ITM). In CoreSight, la frequenza di trace è indipendente da quella del core. Su Cortex-M, la TPIU esporta un clock di trace e 4 pin di data trace.

DWT (Data Watchpoint and Trace Unit)

Si tratta di un'unità in grado di monitorare gli accessi ai dati e il Program Counter della CPU. Contiene 4 comparatori configurabili come hardware watchpoint, ETM trigger, PC sampler o data address sampler. Mediante i comparatori, è possibile fermare la CPU, generare un trigger verso ETM o emettere informazioni di trace selettive su un dato. L'unità implementa inoltre dei contatori, ad esempio sul numero di cicli di clock (CYC), cicli di clock per istruzione (CPI), cicli spesi per istruzioni Load and Store (LSU), oppure numero di cicli spesi in interrupt (EXC). Può emettere anche informazioni sugli eventi di interrupt (entrate e uscite).

ITM (Instrumentation Trace Macrocell)

Emette informazioni di System Trace, sia prodotte dall'unità DWT sia generate dal software in esecuzione. L'ITM appare al software in esecuzione come una periferica mappata in memoria.

Semplicemente scrivendo in opportuni registri (Stimulus Registers) si ottiene l'emissione di informazione sotto forma di pacchetti di trace.

SWV (Serial Wire Viewer)

È una modalità di funzionamento della TPIU tale per cui i dati di trace vengono esportati in modo asincrono attraverso un singolo pin SWO (Serial Wire Output). In questa modalità, l'unità di formattazione interna alla TPIU è disabilitata e quindi non è possibile emettere anche dati di trace provenienti dall'ETM. Il CombiProbe è in grado di acquisire informazioni di System Trace via SWO.

Nella seconda parte dell'articolo saranno mostrati dei casi d'uso reali del sistema formato dalla scheda target e dal debugger, riportando esempi concreti delle prestazioni ottenibili e delle configurazioni necessarie.