[Arm] Why do I get significantly different times when using selective trace (e.g., for task switches)?

2026-02-20 - Comments (0) - Arm

This behavior is caused by the CoreSight trace infrastructure on the chip. Depending on the chip design, several FIFOs exist between trace packet generation and trace collection. If selective trace is configured in such a way that only a small amount of data is generated each time tracing is enabled (for example, a *ContextID* plus a few instructions), then trace packets may remain in a FIFO and be delayed until the FIFO is full. This leads to inaccurate timing.

With **ETM.TimeMode External**, timestamps are applied only when the tool receives the packets. For this reason, internal global timestamping (if supported) is recommended. Note that **CycleAccurate** mode is not suitable in these setups, as it produces only relative cycle information.

In TRACE32, internal timestamps are called **AsyncTimeStamps** (*ETM.TimeMode AsyncTimeStamps*), corresponding to “Global timestamping” in Arm specifications. These timestamps are chip-specific and depend on the implemented ETM. To use them, you must ensure that the timestamp source is running and configure TRACE32 with its frequency, for example:

```
ETM.TImeMode AsyncTimeStamps
ETM.TimeStampCLOCK 50.MHz
```

However, this alone may not be sufficient, as details vary by target system:

- Global timestamping must be supported by the implemented ETM (an optional feature).
- On many systems, you must explicitly enable the timestamp generator. Sometimes this requires setting a single bit; in other cases, it is more complex. On some systems, the source runs automatically after reset.
- You must know the frequency of the timestamp source to configure it with *ETM.TimeStampCLOCK*. This command only informs TRACE32 how to calculate time values; it does not configure the target system itself.

Finally, even with proper setup, it is important to verify how timestamps are positioned by the ETM. Ideally, each data packet should carry its own timestamp. If not, interpolation errors may occur, which can result in values that deviate significantly from real timing depending on the application.