Using DWT and other methods to count executed instructions on Cortex-M.

developer.arm.com/documentation/ka001499/1-0

DEVELOPER DOCUMENTATION

Article ID: KA001499
Applies To: Cortex-M33, Cortex-M4, Cortex-M55, Cortex-M7
Confidentiality: Customer Non-confidential

Summary

What methods are available to help me understand how many instructions have been executed by a Cortex-M processor?

This article was written for Cortex-M3 and Cortex-M4, but the same points apply to Cortex-M7, Cortex-M33 and Cortex-M55. Newer Cortex-M processors at the higher end of performance, such as Cortex-M55, may include an extended Performance Monitoring Unit that provides additional performance measuring capabilities, but these are outside the scope of this article. The smaller Cortex-M processors such as Cortex-M0, Cortex-M0+ and Cortex-M23 do not include the DWT capabilities described here, and, other than the Cortex-M23, do not include ETM instruction trace, but all Cortex-M processors provide the "tarmac" capability for the chip designers.

Answer

The processor contains an optional DWT unit which provides a number of cycle counters.

The basic cycle counter `DWT_CYCCNT` increments on each clock cycle when the processor is not halted in debug state.

A variety of performance monitor counters are provided, which count the number of clock cycles during which the processor diverges from its usual behaviour of executing one instruction per cycle. Most of these performance monitors account for cycles where no additional instruction is executed for one of a number of reasons:

```
DWT_CPICNT - additional cycles required to execute multi-cycle instructions, and
instruction fetch stalls

  DWT_EXCCNT - cycles spent performing exception entry and exit procedures

  DWT_SLEEPCNT - cycles spent sleeping

  DWT_LSUCNT - cycles spent waiting for loads and stores to complete
```

There is also a performance monitor for cycles saved by "folded" instructions:

```
DWT_FOLDCNT - cycles saved by instructions which execute in zero cycles
```

Note that some of these processors are capable of dual-issuing some pairs of instructions. In these cases, two instructions may be executed in one cycle, so the DWT_FOLDCNT may increment rapidly because of the frequent deviations from a rate of one instruction per cycle.

If the processor configuration includes the DWT profiling counters, the instruction count can be calculated as:

```
# instructions executed = DWT_CYCCNT - DWT_CPICNT - DWT_EXCCNT -
                          DWT_SLEEPCNT - DWT_LSUCNT + DWT_FOLDCNT
```

This result is architecturally defined to be approximate. See section "Profiling counter accuracy" in the ARMv7-M Architecture Reference Manual for details.

For a finished, packaged chip, if the chip includes an ETM module for instruction trace, a debugger connected to the trace port output should be able to count instructions exactly, as every instruction is reported in the streaming trace exported on the trace port. However, depending on the processor clock speed and the trace channel bandwidth, it is possible that there may be intermittent gaps in the trace stream due to trace channel capacity overload.

For chip designers who are running a logic simulation of the chip design using the RTL description of the processor, or using the Design Simulation Model (DSM), the exact instruction count can be observed by enabling the "tarmac" logging feature to generate a text log file history of the processor activity during the simulation run, or by enabling the ETM interface of the processor (whether or not the ETM option is implemented in the design) and counting the cycles where the ETMIVALID signal is asserted High.