# Filtered program trace for MCDS Awaiting Agent

- AW Andreas Wikerstal
- **Forum name:** #Trace Analysis

Hello,
in order to increase the trace time window for CPU profiling I am looking into doing on-chip filtering with the help of CTL-scripting. My use case is to measure the task switches on one core but I a would also like to measure more in details on specific functions but a complete program trace.
An example of what I am aiming for is attached in the file. But this does not work I only get the trace of the complete program flow of a core.
Can you please advice on how to set this up?
My target system is a TC399XE.

regards,
Andreas

## Comments (3)

**Houcem Dammak**
1 year ago
Hello Andreas, The provided CTL program doesn't compile. It throws an error "action not allowed outside sequencer states". The action "GOTO level1" in line 3 is outside the state machine. You probably want to add another level. Additionally, you need a statement for selective data trace of the task switches. I'm not sure about your intention when using task_timer. The below CTL program should do the following: */ Trace all write access by core2 to the current task identifier of core2 "TASK.CONFIG(magic:2)". This statement is independent of the state machine. */ When arming the trace "level0" is the active state */ When core2 execute "sieve" entry "level1" becomes the active state. This state transition switches on the Program flow trace and reloads "task_timer" */ As long as "level1" is active "task_timer" keep counting for 200 milliseconds. */ When "task_timer" reach the limit of 200 milliseconds a transition of the state machine to "level2" occur. */ When "level2" state is active, the next execution of "sieve" exit by core2 will stop trace recording and stop the target. //---------- CORE2:: IF Var.Write(TASK.CONFIG(magic:2)) TraceData level0: IF Program(ENTRY:sieve) GOTO level1 level1: IF STATE.ENTER() RELOAD task_timer TraceOn Program IF TRUE() ENABLE task_timer IF TIME(task_timer>=200000.us) GOTO level2 level2: IF Program(RETURN:sieve) TraceTrigger Break //---------- Is this the behavior you want to implement? Or can you give more details? What

TRACE32 software version are you using? You can obtain this using the command: PRINT VERSION.SOFTWARE() Regards, Houcem

AW **Andreas Wikerstal**

1 year ago

Hello, sorry, the failing script is may fault. Made an untested change before sending it to you. The software version: 2023.02.000159199 I have two goals 1. Do performance trace on tasks on a core 2. Do performance trace on functions on a core but I want to limit the performance trace for functions only to run on specific functions. I am doing on-chip trace so I have limited memory (2MB). If I do a full program+data trace on a core the memory fills up in 2.5ms. I want to expand that beacuse I am only interested some functions but I would like to see all tasks. Tried your scripts but I also get "symbol not found" complaining on task.config(magic[2]). I have loaded the orti. Did I missed something else? Adding another variable works e.g. var.write() Thanks for your support. regards, Andreas

**Houcem Dammak**

1 year ago

Hello Andreas, Sorry I made a thinko in my last suggestion. PRACTICE functions couldn't be expanded inside a CTL program. Instead, we need to use the command "Break.ReProgram" with PRACTCE macro expansion "&+" enabled. Example PRACTICE script: //---------- LOCAL &current_task &current_task=TASK.CONFIG(magic:2) Break.ReProgram (&+ CORE2:: IF Write(&current_task) TraceData level0: IF Program(ENTRY:sieve) GOTO level1 level1: IF STATE.ENTER() RELOAD task_timer TraceON Program IF TRUE() ENABLE task_timer IF TIME(task_timer>=200000.us) GOTO level2 level2: IF Program(RETURN:sieve) TraceTrigger Break ) //---------- Please note that, for performance trace on functions, you can also limit the trace to entries and returns of the functions, then you can use the command "Trace.STATistic.AddressDURation" to display statistics about the function execution time. You can use "Trace.Mode Leash" to stop the target when the trace buffer get almost full. Here is an example script: //---------- LOCAL &current_task IF !Analyzer() Trace.METHOD Onchip Trace.Mode Leash MCDS.RESet MCDS.TraceAgents.CLEAR MCDS.TImeMode MCDS CLOCK.ON &current_task=TASK.CONFIG(magic:2) Break.ReProgram (&+ CORE2:: IF Write(&current_task) TraceData IF Program(ENTRY:sieve) TraceEnable Program IF Program(RETURN:sieve) TraceEnable Program ) Go.direct SCREEN.WAIT !STATE.RUN() Trace.STATistic.AddressDURation sieve sYmbol.EXIT(sieve) //---------- Regards, Houcem