How to calculate the CPU load when an OS is used?

2023-12-01 - Comments (0) - OS-aware debugging

The CPU load can be computed based on trace information, i.e. off-chip or on-chip trace is needed in this case.

First, you need to set up OS-aware debugging by loading the OS awareness or ORTI file. Please refer to the TRACE32 manual of the used target operating system (e.g. **rtos_freertos.pdf** for FreeRTOS or **rtos_orti.pdf** for AUTOSAR CP) for more information.

Example 1: Load ORTI file for AUTOSAR CP

```
TASK.ORTI < my_orti_file >
```

Example 2: Load awareness for FreeRTOS on Arm

```
TASK.CONFIG ~~/demo/arm/kernel/freertos/freertos.t32
```

Then, task switches need to be recorded in the trace. Please refer to TRACE32 manual of your target OS for more information.

Example: enable flow trace and task switches recording in the trace based on data trace:

```
Break.Set TASK.CONFIG(magic) /TraceData
```

The CPU load can then be computed as the percentage of time the CPU wasn't in the idle loop. The command **Trace.PROfileChart.TASK** gives such a percentage view. You just have to identify the "idle" task(s).

1. Create a task group for this task with merge or hide option:

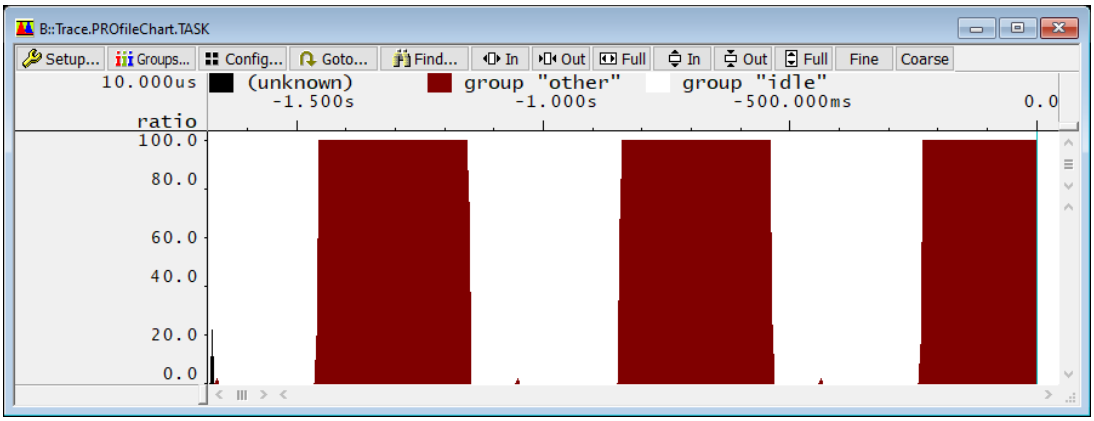
```
GROUP.CreateTASK "idle" "<my_idle_task>" /MERGE
```

e.g.

```
GROUP.CreateTASK "idle" "NO_TASK" /MERGE
```

2. If the chart should not distinguish between the other tasks, you may also merge them:

```
GROUP.COLOR "other" MAROON
GROUP.MERGE "other"
```



Note:

If neither off-chip nor onchip trace is available, then the **SNOOPer** can be used to periodically sample the current task. Please note however that the results are in this case not accurate as when using off-chip or on-chip trace since they are based on periodically collected samples. The target processor needs additionally to support accessing the memory on run-time.

Example:

```
SNOOPer.SELect %Long TASK.CONFIG(magic)
SNOOPer.Arm
WAIT 5.s
SNOOPer.Off
GROUP.CreateTASK "idle" "NO_TASK" /MERGE
GROUP.COLOR "other" MAROON
GROUP.MERGE "other"
SNOOPer.PROfileChart.TASK
```