FAQ 242

Handling Overrun Situations

Keywords
Task overrun; overrun situation; turnaround time; taskCallCount; overrunCount; overrunQueueCount; overrun strategy; ignore overrun; queue task before simulation stop; dSPACE Profiler; first simulation steps; FIRST_SIMSTEP_INCREASEMENT

Question
During real-time simulation the error message Task Overrun: Occurred in task "<task name>", execution step <number> is displayed in ControlDesk or ControlDesk Next Generation. Then the real-time application stops. Why is this error message displayed? How can I suppress the message? How can I analyze the task overrun?

Solution
An overrun occurs when the scheduler attempts to start a task that is still being executed. The sample time might be too small or the call frequency of a task too high. This document summarizes the definitions and methods for avoiding and handling task overruns.

Definition of an Overrun Situation
An overrun situation occurs if a task is requested to start but has not finished its previous execution yet (the predecessor instance of a task has not finished calculating). Refer to the following illustration:

No overrun occurs between the first and second timer interrupts. However, if the T\textsubscript{HW} task, which is triggered by a hardware interrupt block, needs to be calculated, the T\textsubscript{Timer} timer task cannot finish before it is requested to start again. If such an overrun situation occurs, the scheduler detects it and RTI responds according to the overrun settings for the task.
When a task overrun occurs, it is very helpful to analyze the turnaround times of the application tasks. The turnaround time of a task is the time between the task’s trigger and the end of task execution.

The turnaround time comprises:

- The time it takes to execute the functional code of the task
- The time the execution of the task is interrupted by other tasks with higher priorities
- The time it takes to capture variables if the task contains the host service code

The turnaround time does not include the time required to switch from one task to another.

For more information (e.g., an illustration showing the turnaround times for a number of tasks), refer to Overrun Situation and Turnaround Time in the RTI and RTI-MP Implementation Guide.

Analyzing Overrun Situations with ControlDesk or ControlDesk Next Generation

In order to analyze the turnaround times of an application that causes a task overrun, you have to change the overrun strategy from its default value ‘Stop simulation’ to ‘Ignore Overrun’ or ‘Queue task before simulation stop’ via the RTI Task Configuration dialog (Single-processor system: Configuration Parameters dialog; Multiprocessor system: Multiprocessor Setup dialog)

If the overrun strategy ‘Queue task before simulation stop’ or ‘Ignore overrun’ is selected, a task overrun might lead to unexpected simulation results, because tasks are not executed in real time. The application is not real-time-capable when one of these strategies is selected.

You can analyze the overrun situation by means of the following real-time variables. These variables belong to the ‘Task Info’ variable group:

- `turnaroundTime`: To get a continuous display of the turnaround time, you must use a plotter array instrument. Generally, the turnaround time must be smaller than the period of the task (Fixed step size).
- `overrunCount`: The `overrunCount` variable of a task counts the total number of task overruns.
- `overrunQueueCount`: The `overrunQueueCount` variable displays the number of queued task instances and is incremented and decremented if the task strategy ‘Queue task before simulation stop’ is activated.
- `taskCallCount`: The `taskCallCount` variable is incremented with each execution of a task. It does not include the task calls that caused overrun situations.
Analyzing Overrun Situations with the dSPACE Profiler

The dSPACE Profiler is a stand-alone tool for creating a graphical output of the run-time behavior of the real-time application. When investigating run-time problems (e.g., task overruns), the dSPACE Profiler displays the turnaround time distribution of all application tasks. The real-time kernel and its related interventions are shown at the top of the graphical interface. All application task calls and their related turnaround times are displayed below.

When using the Profiler, select at least the logging categories 'Task', 'Task Register', 'Overrun', 'ISR' and 'Idle Mode' (we recommend selecting all categories at first). Furthermore, you can set a trigger condition for the application task that causes the overrun (<Processor board - task name> OD → for overrun detection). The overrun situation is shown and marked with ‘OD’.

For more information about the dSPACE Profiler, refer to its user documentation on the dSPACE Product DVD Tools\Profiler\Profiler_user_doc.pdf.

The latest setup is available under http://www.dspace.com/go/profiler

For MicroAutoBox II, dSPACE Profiler 1.1.5 is required. To check which version is installed, go to the Windows Control Panel and look at the list of installed products.

The dSPACE Profiler is not an official product but was developed as an internal tool.

Preventing Task Overruns

To prevent task overruns, perform one or more of the following steps:

- Increase the sample times of the timer tasks or the fixed step size of the model.
- Reduce the trigger frequency of interrupt-driven tasks.
- Select a different execution mode (single timer task mode or multiple timer task mode) for the model.
- Change the priorities of the tasks.
- Decrease the turnaround time of a task or the execution times of subsystems and blocks by optimizing the modeling inside the Simulink model. For more information, refer to FAQ 211, FAQ 023 or FAQ 292.
- Move parts of your model to timer tasks with slower sample times.
- Find parts of the model which are executed conditionally. Find other methods to take the conditional execution into consideration so that the conditional execution does not increase the turnaround time.
Handling Task Overruns that Occur in the First Simulation Steps

Sometimes you have to handle overruns directly in the model, e.g., if a fixed number of overruns occurs at the beginning of the simulation due to initialization problems. In such cases, a certain number of overruns is acceptable, whereas the application must be aborted in case of further overruns.

For MP systems (as of dSPACE Release 5.2): If the task overrun for the fastest timer task on a CPU of a multiprocessor system occurs only during the first execution step, use the \texttt{FIRST\_SIMSTEP\_INCREMENT} compiler option to avoid the task overrun. For this, call \texttt{DFIRST\_SIMSTEP\_INCREMENT=<Value>} where \texttt{<Value>} is an arbitrary integer ≥ 0.

If the option is set for all the CPUs in the multiprocessor system, \texttt{<Value>} * \texttt{BaseSampleTime} is added to the first simulation step on each CPU. The time available on a CPU for the first simulation step is then as follows: \((\text{StepSizeMultiple} + \texttt{<Value>}) \times \text{BaseSampleTime}\).

In addition to \texttt{FIRST\_SIMSTEP\_INCREMENT}, you can use the \texttt{NUM\_INCREASED\_SIMSTEPS} option to apply the \texttt{FIRST\_SIMSTEP\_INCREMENT} macro to multiple simulation steps on each CPU in your multiprocessor application. For this, call \texttt{-DNUM\_INCREASED\_SIMSTEPS=<Value>} where \texttt{<Value>} is an arbitrary integer ≥ 0.

If the option is set for all CPUs in the multiprocessor system, the increase in simulation step size as defined by the \texttt{FIRST\_SIMSTEP\_INCREMENT} macro is applied to the first \texttt{<Value>} simulation steps on each CPU in your multiprocessor application.

This option is valid only if you specify the same increase value for the first step of each CPU of the multiprocessor system.

If you use this option, the time available for the first simulation step is \((\text{StepSizeMultiple} + \texttt{<Value>}) \times \text{BaseSampleTime}\) (i.e., not real time). During this first simulation step, the timing of I/O signals therefore does not match the timing of the simulation.

The \texttt{NUM\_INCREASED\_SIMSTEPS} option should be used together with the \texttt{FIRST\_SIMSTEP\_INCREMENT} macro.

For SP systems: As of dSPACE Release 2010-B, the above compiler options \texttt{FIRST\_SIMSTEP\_INCREMENT} and \texttt{NUM\_INCREASED\_SIMSTEPS} are also available for SP systems.

Example for SP systems: To give the first three task executions a run time that is twice as long as the model step size, open the Configuration Parameters dialog of your Simulink model (Ctrl+E), navigate to Code Generation > RTI general build options, and type the following command in the 'Compiler options' edit field:

\texttt{-DFIRST\_SIMSTEP\_INCREMENT=1 -DNUM\_INCREASED\_SIMSTEPS=3}

Related dSPACE HelpDesk Documents

- Handling Tasks in the RTI and RTI-MP Implementation Guide
- How to Handle Overrun Situations in the RTI and RTI-MP Implementation Guide
- Handling overruns in multiprocessor systems in the RTI and RTI-MP Implementation Guide.
- RTI Task Configuration Dialog in the RTI and RTI-MP Implementation Reference.
- Build Options Page (CPU Options Dialog) in the RTI and RTI-MP Implementation Reference.

Related FAQs
• **FAQ 211:** Avoiding task overrun errors
• **FAQ 023:** Measuring Execution Times of Blocks and Subsystems
• **FAQ 292:** Measuring Execution Times with the dSPACE Profiler

**FAQ Overview**

http://www.dspace.com/go/faq

**Support**

To request support, please use the form at http://www.dspace.com/go/supportrequest

**Updates and Patches**

Software updates and patches are available at http://www.dspace.com/go/patches. dSPACE strongly recommends to use the most recent patches for your dSPACE installation.

**Important Notice**

This document contains proprietary information that is protected by copyright. All rights are reserved. The document may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the document must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2017 by:

dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

A list of registered dSPACE trademarks is available at:
http://www.dspace.com/go/Trademarks