DDC-I

TECH NOTE

Deos External Clock Synchronization

Synchronizing safety critical systems is often a requirement due to the distributed design and communications requirements of modern avionics applications. A common example is in redundant control systems where a hot standby subsystem is switched into service when a primary subsystem fails. Often, this switchover occurs within hundreds of microseconds and requires that the hot standby be working with the same data set and synchronized to the same control loops as the primary subsystem's data. The degree of drift or jitter on the system clock between the primary and standby systems can vary so they require a mechanism to quickly and easily synchronize time. Additionally, some safety critical systems require that all subsystems synchronize before they start executing, while other systems may allow subsystem controllers to operate out of sync when they first power-up and then sync up later. Fortunately, the Deos™ Real-time Operating System has several mechanisms to address these different synchronization requirements.

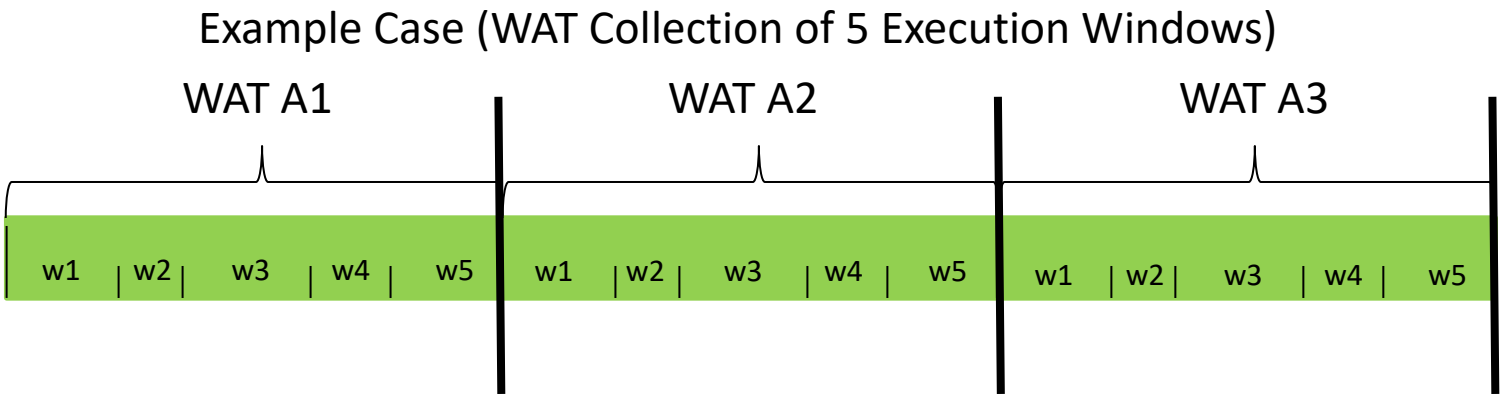
For most systems needing synchronization the simplest approach in a Deos based system is clock adjustment managed at the application level; where the developer can shift the major time frame clock ahead or behind, or a little bit at each time. By performing a minor clock shift over a series of cycles, the developer can align the major time frame clock to start where it's needed with minimum impact to application time lines. This paper describes how this approach is done.

Key Features Overview

- An Approach to Synchronize a Deos System
  - Applicable for Redundant and Coordinated Control System Designs
  - Leverages Deos External Clock Synchronization Feature
- Application Level Implementation
  - Triggering can be Through PTP IEEE-1558, Deterministic Communications, or Other
  - Doesn't Require Specialized Clocking Hardware or Lower Level Software Support
- Incremental Clock Shift
  - Adjusts Clock Drift with Minimal Impact on Timeline Operation

Time partitioning is essential for safety/mission critical systems. Typical embedded designs use on-chip processor clocks for this partitioning. Recent designs have shown an increased use of external clocks to provide system timing synchronization. For example, external clock sources have been implemented in Time Triggered Ethernet (TTE) and FPGAs. The Deos External Clock Synchronization feature allows a properly integrated user application to adjust the execution timeline, managed by the on-chip processor clocks, to align with an external clock. This is accomplished with minimal time partitioning interference.

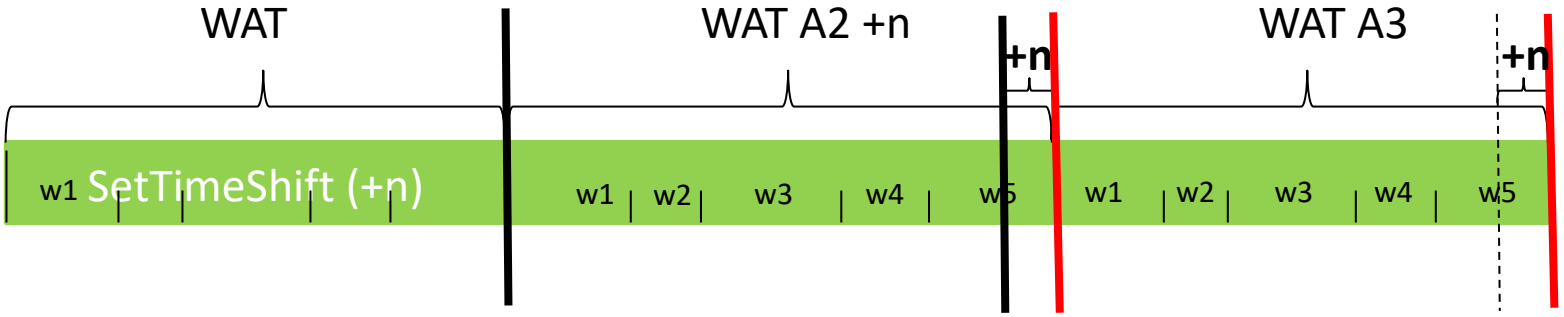
In the Deos environment, the Deos Kernel manages a collection of one or more execution windows. The collection is called a Windows Activation Table (WAT) and repeats across the timeline as execution proceeds. The WAT is synonymous with the hyper-period in Deos rate monotonic analysis (RMA) scheduling and the major frame in Deos ARINC-653 scheduling.



For the example case, WAT A consists of 5 execution windows. The diagram shows the WAT repeating 3 times in the timeline. The solid black vertical line represents the Deos Kernel scheduling point for the WAT, also known as the WAT boundary.

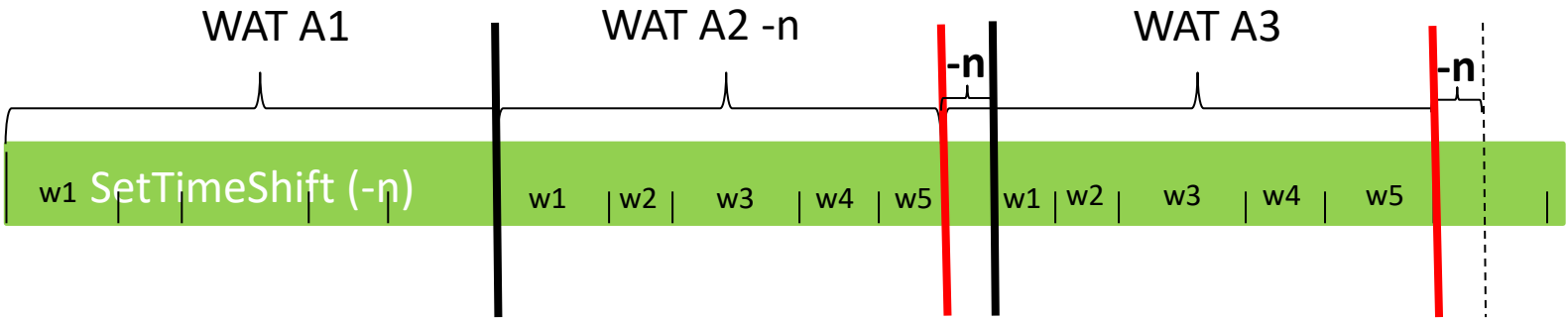
All applications in the Deos environment can determine their timing in the WAT through a Deos Kernel API. Additional integration features can be used to allow an application to obtain time data from an external clock source. The application can calculate an offset between its time in the WAT and the external source. Through an additional API, the application can use this offset to request a shift in the WAT boundary to align with the external clock source. The Deos Kernel, managing the WAT, and the Deos platform abstraction library (PAL), managing the on-chip clocks, can apply that offset to the next WAT. The following diagrams depict the plus and minus offset cases.

+n Case (WAT Collection of 5 Execution Windows)



Any application in the WAT, given knowledge of the external clock, can be used to determine the required shift time. Since time partitioning is affected, the application requesting the shift must have the highest design assurance level in the system. In the diagram, the application executing in window 1 (w1) has the necessary timing knowledge to determine the offset between the on-chip processor clocks and the external clock source. The application calls the ***setTimeShift (+n)*** API with the required offset. The Deos Kernel and the Deos PAL apply the offset at the WAT boundary between WAT A1 and WAT A2. WAT A2 is extended by time n. Note that this extends window 5 (w5) to set up a new WAT boundary in red. WAT A3 and all subsequent WATs return to their original timing.

–n Case (WAT Collection of 5 Execution



A negative offset works in a similar manner. Again, the application in window 1 (w1) makes the offset determination and calls the ***setTimeShift (-n)*** API. The Deos Kernel and the Deos PAL apply the offset at the WAT boundary between WAT A1 and WAT A2 with WAT A2 shortened by time n. Window 5 (w5) is shortened to set up a new WAT boundary in red. WAT A3 and all subsequent WATs return to their original timing.

With minimal time partitioning effect, the Deos External Clock Synchronization feature gives applications execution timeline control to align on-chip processor clocks with external clock sources.

Please contact DDC-I for more information on this and other available Deos synchronization mechanisms to fit your system requirements and design.