

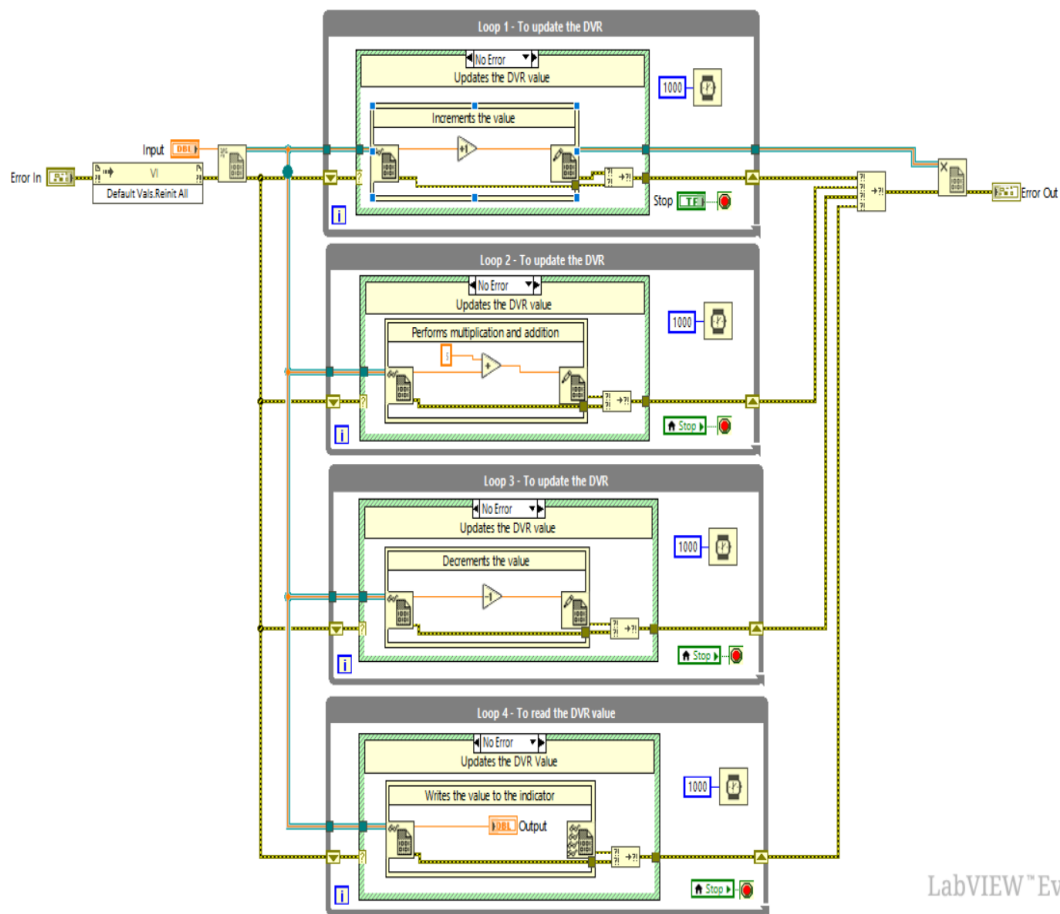
DVR Access

Create a VI that creates a DVR and passes it to 4 loops. In 3 loops, perform some updates of the DVR, and in 4th loop, display the DVR data in an indicator. Modify the wait times and other settings in the In Place Structure and document the behaviors.

Items to play with:

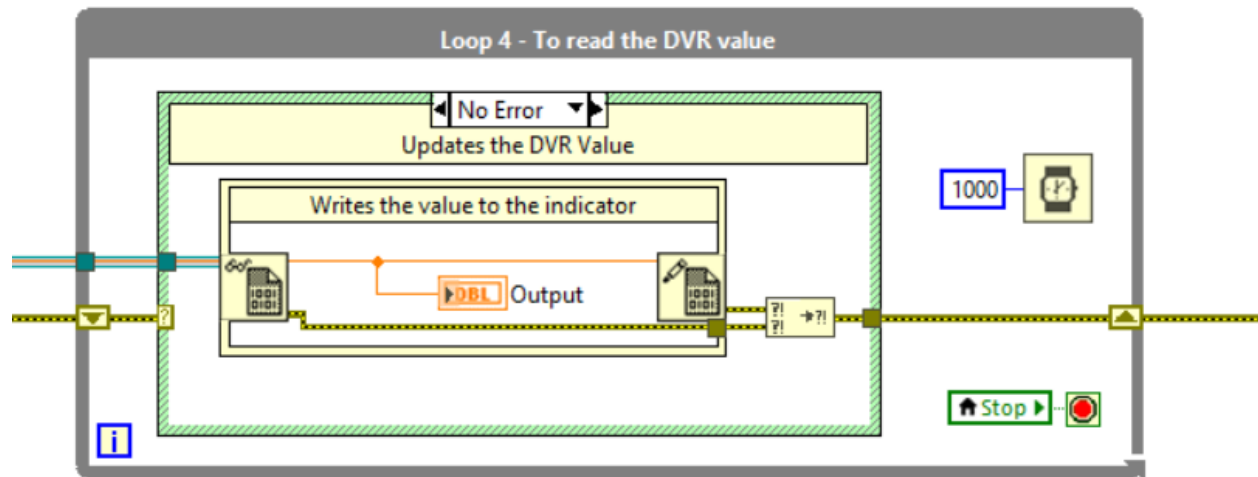
1. Wait time/delays in the loops
2. “Mark as Modifier” in Input terminal
3. “Allow Parallel Read-only Access” in output terminal

VI Block diagram:



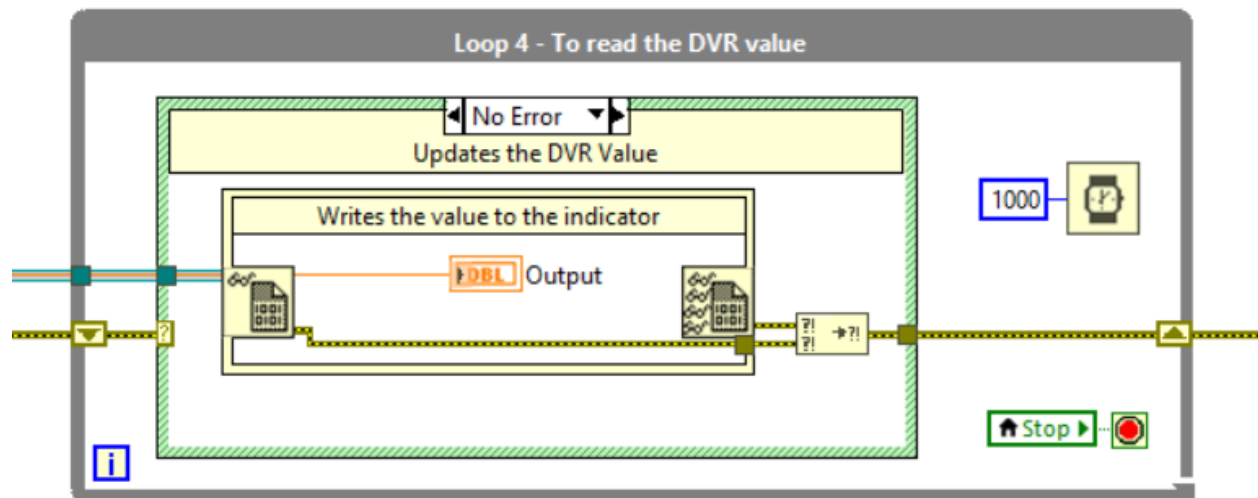
Same Wait time, Mark as modifier off and “Allow Parallel Read-only Access” off

All the loops get executed, but not in a specific order. The data gets updated in the indicator every second. Loop 4 also must wait like all other loops to read the data.



Condition 2:

Same Wait time, Mark as modifier off and “Allow Parallel Read-only Access” On



When Allow parallel Read only access is turned on for the right node in Loop 4, it writes the value to the indicator more frequently as it does not wait like all other loops to get the access.

Condition 3:

Different Wait time, Mark as modifier off and “Allow Parallel Read-only Access” off

- Loop 1: 1000(ms)
- Loop 2: 2000(ms)
- Loop 3: 500(ms)
- Loop 4: 100(ms)

Loops 1 and 2 update the DVR value for one and two seconds respectively, whereas Loop 3 updates frequently for 500 ms.

Loop 4, being the most frequently running loop for 100 ms once, it is **real-time** and fetches the DVR data without any delays.

Conversely, when Loop 4 is assigned with potentially greater wait time, the data displayed is not real-time.

Mark	as	Modifier	On
------	----	----------	----

It is a way of letting know LabVIEW that the data is undergoing a modification. This optimizes the performance of the memory. It does not show visible variations in the above assignment.