ARTIQ
Progress report

July 4th, 2016

1 Release 1.1

We have released ARTIQ 1.1. It is a bugfix release and we recommend that all 1.0 users upgrade.

2 Improved multiple scan support (2.0)

We have added a class `artiq.language.scan.MultiScanManager` that makes it easier to write experiments containing a large number of potentially scanned variables. The class is an alternative to the deep nesting of `for` loop.

Scan widgets in the GUI now have a button to disable (set to "no scan") all the other scans in the experiment.

3 Batched dataset mutation (1.1)

The `mutate_dataset` function can now accept a range of indices (passed as tuples), in order to spread the per-call overhead of a dataset mutation across multiple values.

4 Seamless handover across experiments (2.0)

The last position of the time cursor (`now`) is now saved and restored across different experiments. Core device resets (that involve flushing RTIO FIFOs and setting the time cursor shortly after the current value of the RTIO counter) are handled explicitly by a method `reset()` of the core device driver.

The idle kernel is still started whenever there is no connection with the control PC. It should check by itself whether it should do something using a prologue such as the following:

```
start_time = now_mu() + seconds_to_mu(500*ms)
while self.core.get_rtio_counter_mu() < start_time:
    pass
self.core.reset()
```

5 Pause checks (2.0)

We have added a `check_pause()` method to the scheduler, which returns `True` if there is a condition that could make `pause()` not return immediately (termination requested or higher priority run).

The typical purpose of this function is to check from a kernel whether returning control to the host and pausing would have an effect, in order to avoid the cost of switching kernels in the common case where `pause()` does nothing.

6 Logging improvements (2.0)

Log messages are now stored locally by the dashboard (instead of the master with a buffer synchronization mechanism for the clients). The master simply broadcasts new logging messages as they are generated, without storing them (except into the logging files, which remain supported as before). Filtering is now performed on each new message upon reception.

The dashboard also now adds its own messages to the logging docks.

7 Compiler (2.0)

We've added support for Python modules, dynamically created user classes (to support lists of scan points of different types, i.e. multiple simultaneous managed scans on the core device), and `min()` and `max()` builtins.

8 Browser (2.0)

The browser can now upload its datasets to the master. Aspects of the HDF5 experiment argument loading have been refined.

9 Bug fixes and minor features

We have implemented many other small features and fixed a number of bugs into ARTIQ. See the Git logs for details.

10 Site visit

The NIST site visit took place June 7th to 10th. The timing tutorial was particularly well received.

11 Hardware designs

11.1 Sayma DSP gateware

Further design work has gone into the DSP channel design for the ARTIQ Sayma cards. From preliminary tests four fully independent two-tone channels at up to 1.25 GHz sampling rate appear to be feasible.

11.2 DRTIO

We have successfully configured the Si5324 on the KC705 to perform jitter filtering of the clock recovered by the Kintex-7 transceivers.