CBE Status
for
Prototype Board Testing

Output products

- Raw lag frames in ASCII format to file.
- Lag frame sets after CBE integration.
- Something else for the RTDD (???).

Simplified CBE

Design of CBE simplified for purposes of prototype board testing.

- Single node; no monitor or control processes.
- Command line interface, supplies configuration to be provided ultimately by control process.
- Configuration for prototype board testing through two XML files: one for configuration of shared memory owned by the "controlling process" (*i.e*, you, for board testing; control process, ultimately), one to provide configuration or processing instructions.

Running the CBE

- Example: `input --source cbe:10000 --memconfig mc.xml obs.xml`
- Times out after approximately two backend integration periods without incoming frames (after first frames of an "observation" have arrived.)
- Simple simulator, `baseline_sim`, exists. Produces frames (without much variation in content) at a rate and to a destination given by the user.

“Obs” file

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing>
    <!--
    <fn_sequence object="lag_frame">
      <fn name="ascii_dump"/>
    </fn_sequence>
    -->
    <fn_sequence object="lag_set">
      <fn name="bartlett_window"/>
      <fn name="fft"/>
    </fn_sequence>
    <fn_sequence object="integrated_vector">
      <fn name="ascii_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

obs element

```
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing/>
</obs>
```

The values of `correlator_dump_period` and `integration_period` are expressed as number of periods of the 128 MHz correlator clock.

data_processing element

- single precision floating point

```
<data_processing precision="single">
  <fn_sequence/>
</data_processing>
```

- double precision floating point

```
<data_processing precision="double">
  <fn_sequence/>
</data_processing>
```

fn_sequence element

- applied to lag frame

```
<fn_sequence object="lag_frame">
  <fn name="ascii_dump"/>
</fn_sequence>
```

- applied to lag set

```
<fn_sequence object="lag_set">
  <fn name="hann_window"/>
  <fn name="fft"/>
</fn_sequence>
```

- applied to integrated vector

```
<fn_sequence object="integrated_vector">
  <fn name="ascii_output"/>
</fn_sequence>
```

Raw lag frame output example

```
<?xml version="1.0" encoding="utf-8"?>
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing>
    <fn_sequence object="lag_frame">
      <fn name="ascii_dump"/>
    </fn_sequence>
</obs>
```

Output consists of ASCII files, one for each CCC number seen in data stream (names: ccc-<num>.txt).

Integrated lag set output example

```
<?xml version="1.0" encoding="utf-8"?>
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing>
    <fn_sequence object="integrated_vector">
      <fn name="ascii_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

Output consists of ASCII files, one for each integration (names: <obs name>-<timestamp>.txt).

FFT examples

- applied to individual lag sets

```
<obs name="t1" epoch="0" start="0"
     correlator_dump_period="1200" integration_period="120000000">
  <data_processing>
    <fn_sequence object="lag_set">
      <fn name="fft"/>
    </fn_sequence>
    <fn_sequence object="integrated_vector">
      <fn name="ascii_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

- applied to integrated lag set

```
<obs name="t1" epoch="0" start="0"
     correlator_dump_period="1200" integration_period="120000000">
  <data_processing>
    <fn_sequence object="integrated_vector">
      <fn name="fft"/>
      <fn name="ascii_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

Windowing example

```
<?xml version="1.0" encoding="utf-8"?>
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing>
    <fn_sequence object="lag_set">
      <fn name="bartlett_window"/>
      <fn name="fft"/>
    </fn_sequence>
    <fn_sequence object="integrated_vector">
      <fn name="ascii_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

RTDD output

Not yet implemented!

```
<obs name="t1"
     epoch="0"
     start="0"
     correlator_dump_period="1200"
     integration_period="120000000">
  <data_processing>
    <fn_sequence object="lag_set">
      <fn name="bartlett_window"/>
      <fn name="fft"/>
    </fn_sequence>
    <fn_sequence object="integrated_vector">
      <fn name="rtdd_output"/>
    </fn_sequence>
  </data_processing>
</obs>
```

Unresolved issues

- Framework to assemble lag sets from lag frames has been implemented, but no lag set assembly is yet being done by CBE processes. Primary stumbling block is how to represent lag set "assembly instructions."
- Construction of lag sets from 7–bit correlation products.

More detail in *Correlator Backend and Fast Formatter* presentation.