
Subject: Re: problem with a processor (software-wise)

Posted by [miree](#) on Mon, 11 Aug 2014 11:48:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Tayfun,

this post is related to the error message Assertion `input_arrays_[channel].size() > i' failed.

If you want to write processors with arrays, you need to know the structure of an Array. This is explained in the tutorial.pdf in Section 4.2 "Elder Arrays: Writing a DSSSD Processor". I'll try to point out the relevant thing:

Arrays are composed of N entries, where N can be different for each event. If an array is defined by

```
NAME_INPUT_ARRAY(my_array, 4) // 4 different indices allowed: 0,1,2,3
```

It can look like this: (input_array_size(my_array) == 4)

```
    |entry0 | entry1 | entry2 | entry3 |
-----
index | 1   | 3   | 2   | 3   |
value | 3.2 | 100.5 | 3.3 | 234.5 |
-----
```

Or it can look like this: (input_array_size(my_array) == 6)

```
    |entry0 | entry1 | entry2 | entry3 | entry4 | entry5 |
-----
index | 3   | 0   | 1   | 2   | 2   | 1   |
value | 3.2 | 100.5 | 3.3 | 234.5 | 150.1 | 3.14 |
-----
```

Or it can be empty: (input_array_size(my_array) == 0)

```
    |
-----
index |
value |
-----
```

The index field will only contain numbers 0,1,2 or 3. It can have an arbitrary number of entries. Each index may come zero or one or many times. Each entry has one value.

The meaning of the index depends on the context. It can be the channel of MultihitTDC, the crystalID of a HPGe-ARRAY, or the LYCCA-module number.

The correct way to loop over an array is like this:

```
int N = input_array_size(my_array); // N can be any positive integer or zero
for (int i = 0; i < N; ++i)
```

```

{
  int  index = input_array_index(my_array, i); // index refers to the index of the i-th array entry
  double value = input_array_value(my_array, i); // value refers to the value of the i-th array
entry

  // do something with the value and the index
  // .. for example testing
}

```

The assertion failure is caused by accessing an entry that is beyond the size of the array.
Example: If you get an array with 3 entries and you call entry 5, it will not work.

```
input_array_index(my_array, 5); // this will cause an assertion failure if the number of entries is
smaller than 6
```

That's it

There are a few traps that can happen if you are new to that. Looping over an array is one of them.

So don't hesitate to ask.

Also, there is a possibility to make the condition test by calling the function 'condition_valid'

```

for( //...
{
  int  index = input_array_index(my_array, i);
  double value = input_array_value(my_array, i);
  //...

  // the following works if 'reference_gate' is defined
  // with 'NAME_CONDITION_ARRAY(reference_gate, Window1D,
parameter(num_channels))'
  if (condition_valid(value, reference_gate, index))
  {
    //...
  }
}
}

```

Best regards,
Michael
