LATC interface notes

Introduction

LATC as a system doesn't provide an interface for manipulating the LATC xml as a system. Also, due to some features of the LATC compiler (LATC_parser), there are subtleties in attempting to simply "chain" different sets of XML together to create a correct description of the detector

Interface

The <foo> package contains a simple interface to an in-memory model of the LATC register space. A top level object, LATC_LAT, is provided which constitutes the top level interface to the LATC register hierarchy. This package contains several base classes: LATCComponent, LATCComponentDict, LATCRegister. The top level object is instantiated with a simple:

```
lat = LATC_LAT()
```

LATCComponent

The base class of all LATC component objects (such as LATC_LAT, LATC_TEM, LATC_AFE). LATCComponent objects contain sets of LATCComponents and sets of LATCRegisters. These sets may be empty.

All LATCComponent objects define the following methods:

```
name()          Returns the component's name (a string)
components()    Returns a dict of LATCComponent objects (may be empty).
                  Example:  lat.components().keys() returns:
                  ['AEM', 'TEM', 'GEM']
nComponents()   Returns the length of the components() dict
registers()     Returns a dict of LATCRegister objects (may be empty).
                  Example: lat.components()['AEM'].registers().keys() returns:
                  ['trgseq', 'aem_configuration']
addComponent(name, component)    Adds a component to the object.
                                 ***Not normally used by the user.
addRegister(name, register)      Adds a register to the object.
                                 ***Not normally used by the user.
```

There also exists a more intuitive interface for LATCComponents. The user can reference the components and registers of a component by name. For example, the following two code snippets are equivalent:

```
>>> aem = lat.components()['AEM']
>>> trgseq = lat.components()['AEM'].registers()['trgseq']
```

```
>>> aem = lat.AEM
>>> trgseq = lat.AEM.trgseq
```

LATCComponentDict

A LATCComponentDict is a dictionary of LATCComponent objects. It inherits from the python dict class. It is used in component() dictionaries as a collection of substantively identical objects, mainly to ease the referencing of these objects.

```
>>> lat.TEM.keys()
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
>>> lat.TEM[5]
<latcBuilt.LATC_TEM object at 0x15caa30>
```

LATCRegister

A LATCRegister is a BitField with an integrated, unchangable field definition. It is constructed with a dictionary indicating the bit field names and positions. For example, the following code creates a LATCRegister with a name of engine_4, initialized to 0, with a set of bitfields spanning bits 0 through 29:

```
>>> engine4 = LATCRegister('engine_4', 0x0, {'prescale':(0,8),
                                             'inhibit':(15,16),
                                             'calstrobe':(16,17),
                                             'tack':(19,20),
                                             'four_range':(20,21),
                                             'zero_suppress':(21,22),
                                             'marker':(22,25),
                                             'destination':(25,29) })
```

LATCRegisters implement the following methods:

```
name()      Return the name of the register
set(ui)     Set the value of this register to ui
bf_def()    Return the bitfield definition dictionary, as above
```

Fields of a LATCRegister can be referenced in the same way as components of a LATCComponent:

```
>>> p = engine4.prescale
```