Fixed-Point Bit Vector

Introduction:

Using the MyHDL did speed up development for digital code. However, within our organization the MyHDL environment is mainly used by system level people and digital engineers. Different requirements of using the same simulator environment, i.e. system architects to quickly model a system and digital engineers having a cycle accurate model let to the idea, (that have already be posted before on the MyHDL website) to introduce a fixed point bitvector type, i.e. fixbv. To test this new feature we introduced a new class, i.e. fixbv. This class defines the fixed point bit vector. The starting-point is the intbv class for the version 0.9 MyHDL code. Basically, the fixbv is an extension from the intbv. It contains the same features with some additions features from which the main addition is the 'shift' attribute. Basically the notation of a fixbv number is defined as:

$$value = VAL \cdot 2^{SHIFT}$$

fixbv class

```
def __init__(self, val, shift, min=None, max=None, _nrbits=0):
```

As mentioned, the same structure as intbv is used, with an additional mandatory parameter 'shift'.

Basic assumption: all internal values and parameters are stored as integer, so the 'canonical' form uses all integer values for val, shift, min, max and _nrbits. For convenience val, max and min can also be passed as float. In that case they are internally converted to integer (based on shift)

Initializing a new fixbv can be done in multiply ways, i.e.:

`A = fixbv(val=<float>, shift=<int>)`
Initialize the fix point bitvector with a 'float'. The float value is stored internally to the nearest fix-point value. The calculation is done as follow: $$\_val = \lfloor val \cdot 2^{-shift} + 0.5 \rfloor$$ $$\_shift = shift$$ $$\_min = none$$ $$\_max = none$$

`A = fixbv(val=<float>, shift=<int>, min=<float>, max<float>)`
Initialize the fix point bitvector with a 'float' and limit the range by specifying the ranges with floating point limits. The float value is stored internally to the nearest fix-point value. The calculation is done as follow: $$\_val = \lfloor val \cdot 2^{-shift} + 0.5 \rfloor$$ $$\_shift = shift$$ $$\_min = \lfloor min \cdot 2^{-shift} + 0.5 \rfloor$$ $$\_max = \lfloor max \cdot 2^{-shift} + 0.5 \rfloor$$

`A = fixbv(val=<int>, shift=<int>)`
Initialize the fix point bitvector with a 'int'. The integer value is stored one-to-one in the internal _val attribute. The assignment is done as follow: $\_val = val$ $\_shift = shift$ $\_min = none$ $\_max = none$

`A = fixbv(val=<int>, shift=<int>)`
Initialize the fix point bitvector with a 'int'. The integer value is stored one-to-one in the internal _val attribute. The assignment is done as follow: $\_val = val$ $\_shift = shift$ $\_min = none$ $\_max = none$

`A = fixbv(val=<int>, shift=<int>, min=<int>, max=<int>)`
Initialize the fix point bitvector with a 'int'. The integer value is stored one-to-one in the internal _val attribute. The assignment is done as follow: $\_val = val$ $\_shift = shift$ $\_min = min$ $\_max = max$

Internally the fixbv class does contain the same re-implemented functions as defined in the intbv, additionally an align-function has been added. The align-function is a member of the fixbv class, aligning input data in the following order:

<fixbv A>.align(<fixbv B>)
The return value is calculated using the following equation: $value = B.\_val \cdot 2^{(B.\_shift - A.\_shift)}$

<fixbv A>.align(<intbv B>)
The return value is the _val of the class B $value = B.\_val$

<fixbv A>.align(<float B>)
The return value is calculated using the following equation: $value = \lfloor B \cdot 2^{-A.\_shift} + 0.5 \rfloor$

<fixbv A>.align(<int B>)
The return value: $$value = B$$

Conversion from 'fixbv' to ...

The fixbv class can be converted when assigned to another data type. The following proposal is currently implemented.

<int A> = int(<fixbv B>)
The assignment is done as follows: $$A = B.\_val$$

<float A> = float(<fixbv B>)
The assignment is done as follows: $$A = B.\_val \cdot 2^{B.\_shift}$$

repr(<fixbv B>)
The assignment is done as follows: $$string = repr\left(fixbv(4, -8)\right)$$ $$string = "fixbv(4, -8)"$$
str(<fixbv B>)
The assignment is done as follows: $$string = str\left(fixbv(4, -8)\right)$$ $$string = '0.015625'$$