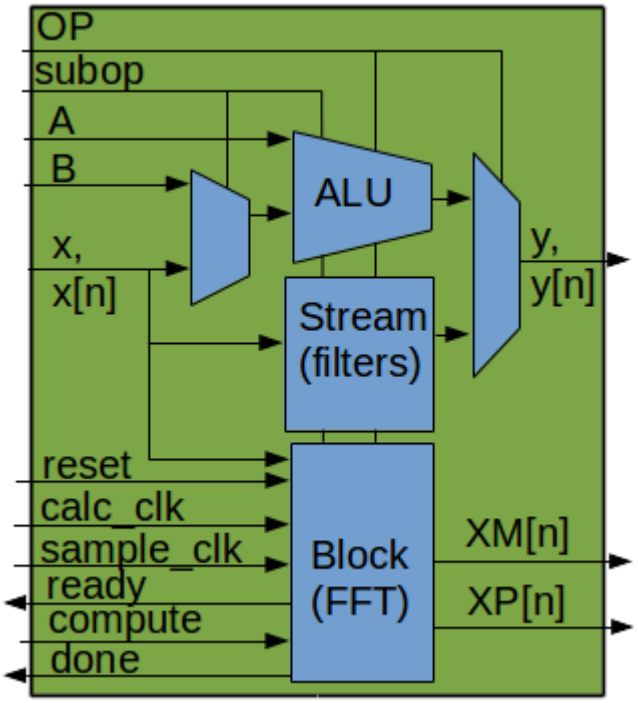
A Virtual DSP testbench

http://www.yanthia.com/online/projlets/VirtualDSPtestbench/index.html

There already are a number of SW tools available for testing digital filter components. One can also just write the code and drop it into an FPGA.
But just for the fun of it I thought I'd see what I could get out of UVM and GNUplot with just a few hours work. This means creating a virtual signal generator, and a virtual oscilloscope & spectrum analyzer. I used GNUplot for the latter, but I'll not elaborate on that part of the setup.
The signal generator part is shown below where *UVM_TESTNAME* specifies the type of DUT, and the input signal type and parameters are passed on the command line as *+uvm_set_config_ints*.

The testbench has a "multi-purpose" DUT where one can drop in any of three types of components (see box to the right): two-input "ALU" types, and single-input "stream" types like digital filters, and "block" types like FFTs that require some amount of data to be collected before an operation can begin.

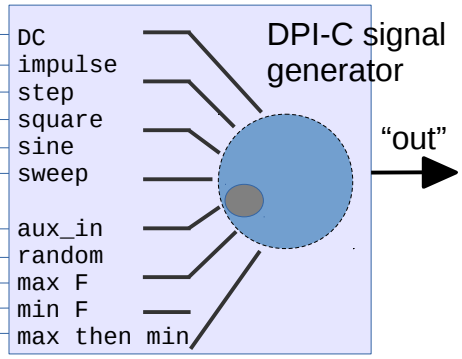
This is about as easy as it gets. Once the RTL is done it's only minutes to drop it into this testbench and get results.




```
+UVM_TESTNAME=OP_LPFfir1_test

  +uvm_set_config_int=*,SIGNAL,0  (DC)
  +uvm_set_config_int=*,SIGNAL,1  (IMPULSE)
  +uvm_set_config_int=*,SIGNAL,2  (STEP)
  +uvm_set_config_int=*,SIGNAL,3  (SQUARE)
  +uvm_set_config_int=*,SIGNAL,4  (SINE)
  +uvm_set_config_int=*,SIGNAL,5  (SWEEP)

  +uvm_set_config_int=*,SIGNAL,7  (FILE_READ)
  +uvm_set_config_int=*,SIGNAL,8  (RANDOM)
  +uvm_set_config_int=*,SIGNAL,9  (MINSAMPLES, 4 samples/cycle)
  +uvm_set_config_int=*,SIGNAL,10 (MAXSAMPLES, 1024 sample sine wave)
  +uvm_set_config_int=*,SIGNAL,11 (MIN_MAX, burst of '9' then '10')
```



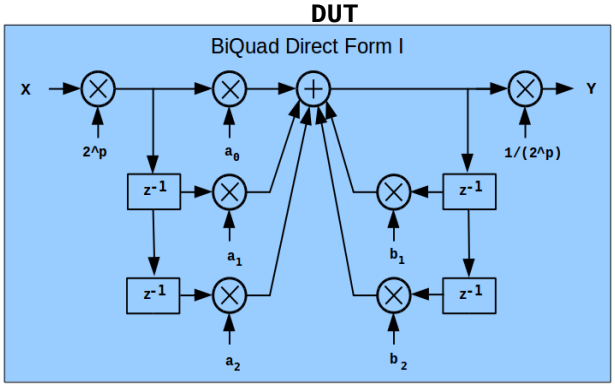
Here's an example:
+UVM_TESTNAME=OP_BiQuadLPF_1_test +uvm_set_config_int=,SIGNAL,7*
where the aux_in is the "GoldfishBubble" audio file and the biQuad LPF implements

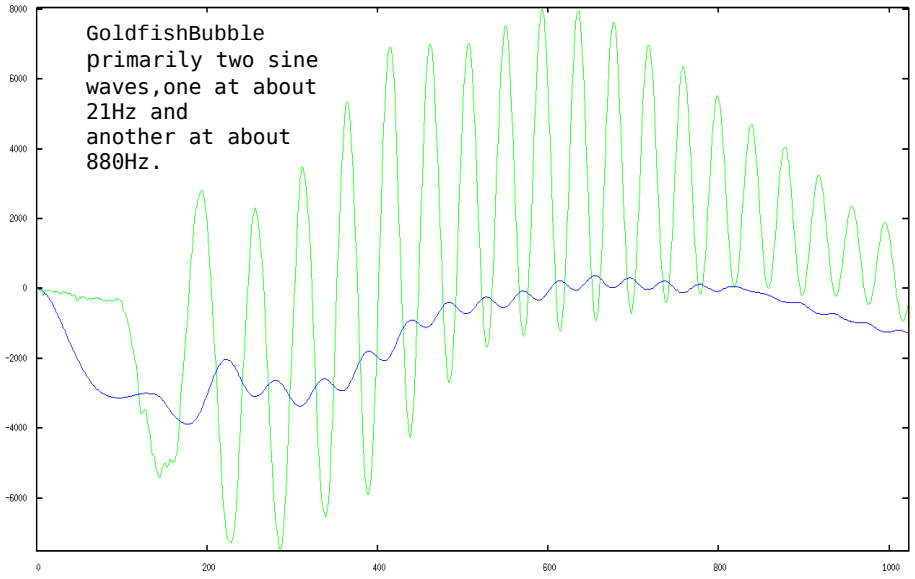
$$y[n] = (2^P * (a0*x[n] + a1*x[n-1] + a2*x[n-2] + (-b1)*y[n-1] + (-b2)*y[n-2]) /(2^P))$$

This is a fixed point implementation with a minimum 1024 bits resolution. Constants are given below on the left. The input is shown in green and output is shown in blue with the 880Hz component greatly attenuated:

LPF	a0 =	0.00019897136836160
Fs = 44100	a1 =	0.00039794273672320
Fc = 200	a2 =	0.00019897136836160
Q = 0.707	b1 =	1.9597066626643 (note sign change)
	b2 =	-0.960502548137 (note sign change)

DUT





Dan Downs OstensiblyMyID@gmail.com Scottsdale, AZ, November 2016