Virtual Test with VHDL-AMS For a Generator of Analog and Mixed Signal Virtual Components

Belgacem BABBA, Gauthier BARRET, Frederic POULLET
Dolphin Integration, 39 Av. du Granier, BP 65 - ZIRST 38242 Meylan-FRANCE
Phone: (33) 4.76.41.10.96, e-mail: circuitware@dolphin.fr

Abstract

With the increase in the complexity of systems with mixed-signal components, many systems designers recognize advantages and necessity of the mixed-signal simulation of these systems as a whole. Unfortunately, modeling and simulating such complex systems takes an enormous amount of time. The first problem identified is the speed of analog simulation. The second problem is how is difficult to write accurate and rapid models of a complex analog and mixed-signal system. Mixed-signal EDA tools has arguably made significant improvement over the past two years. Finally with the Analog and Mixed Signal HDL standard (VHDL-AMS), accurate and rapid model of a complex electronic system is made possible.

This paper presents a case study; the Virtual Test of a virtual component (VC) "ADMIR", a flexible analog to digital converter. The power of VHDL-AMS language makes it possible to write models with different abstraction level of the test chip and the "testboard" of ADMIR..

1. Introduction

The "Virtual Test" concept can improve significantly the test program development. This is extremely valuable to an organization, due to the high cost of the hardware testers. The emergence of Virtual Test had been delayed until now due to many factors, including the complex mixed-signal design space, the lack of standardized HDLs for behavioral level design and inadequate design tools. Another application of the Virtual Test is to create an analog testbench, a simulation environment that can be used during the design process of a component as the reference validation system.

All aspects of the tester and the device under test must be accurately modeled in the Virtual Test environment. The main problem of the model creation is that the model must be detailed enough to behave as if on the real tester, but not so detailed that the simulation time is unacceptable. Adding to this problem is the need to work on both analog and digital components at the same time.

With VHDL-AMS, the model writer can focus on the model equations without the distraction of simulator data structures. In traditional circuit simulators, the current model equations and simulator specific implementation details have been intermixed. This makes it very difficult to add new models to such simulators as both the model equations and the simulator internals must be completely understood. With VHDL-AMS, the model writer does not have to worry about simulator internals, only about the current model equations. These encourage designers to create models with different abstraction level that improve significantly the simulation time.

This paper presents the Virtual Test of an analog to digital converter the virtual component ADMIR. The paper is structured as follows. In section 1, a summary of the main features of the flexible analog to digital converter (ADMIR) and the "testboard" used for test are described. In section 2, some VHDL-AMS models of the device under test and the "testboard" are discussed. Section 3, a presents the design methodology, a summary of the benefits of the proposed technique and an attempt to define future work required to achieve Virtual-Test as a commercial reality.

2. ADMIR, a generator of flexible ADCs.

2.1. Overview of ADMIR

ADMIR is a generator of programmable incremental ADCs. The ADCs that can be created by the generator are high precision ADCs for instrumentation and measurement applications, with an equivalent resolution between 10 and 16 bits, and with an output data frequency (Fs) up to 50 kHz.

The overall working area (Differential non linearity "DNL", Integral non linearity "INL", Fs) covered by the generator is the following one:

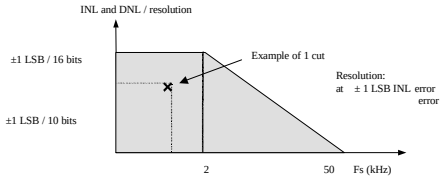

Figure 1. ADMIR working area

The generator is a software program used to create cuts inside the working area with an INL and DNL lower than 1 LSB. The generator can create all the views and data that are needed by the user to implement the ADC in the final chip. The cut generated is a mixed-mode virtual component. It is a programmable ADC, containing an analog part and a logical part.

The analog part contains a modulator, an analog multiplexer and an internal voltage reference. The analog modulator is either a first order, or a second order, or a second order with calibration. In this paper we will focus on one type of modulator: the first order incremental ADC.

The logical part depends on the parameters set by the user (the number of channels, resolution...). Different possibility of parameters values for the digital part, result in a high number of possible instances.

The first order incremental converter is an efficient solution for accuracy varying from 10 to 16 bits, and for an input signal frequency varying from DC to some hundreds of Hertz.

2.2. Conversion principle

The incremental conversion is based on first-order ΣΔ Modulator; the main difference is that the integrator and the counter are both reset before each conversion. As a result, the digital part is much simpler than in a ΣΔ Converter, it can convert DC signals and doesn't exhibit an error spur at specific signal levels. On the other hand, it is slower than an oversampling ΣΔ Converter.

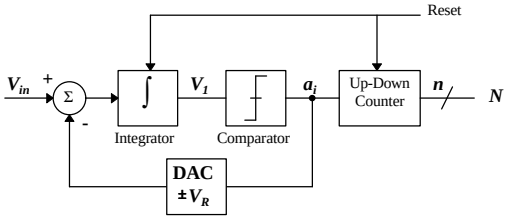

Figure 2. 1st order SD Modulator.

The modulator of ADMIR is composed of a switched-capacitor integrator, a comparator and switch control logic. Conversion is made periodically on $\boldsymbol{p}$ cycles, each cycle is divided into four-phases. The integrator output voltage is designated by $\boldsymbol{V_{1(i,j)}}$, and the comparator output value is designated by $\boldsymbol{a_{i,j}}$ where $\boldsymbol{i}$ corresponds to the integration cycle number $\boldsymbol{i}$, and $\boldsymbol{j}$ corresponds to the clock phase number $\boldsymbol{j}$.

The purpose of this paper is not to explain precisely the principles of the incremental conversion technique, we will give only the equations representing the functionality, which are interesting to understand the VHDL-AMS models which will be described in paragraph 2.

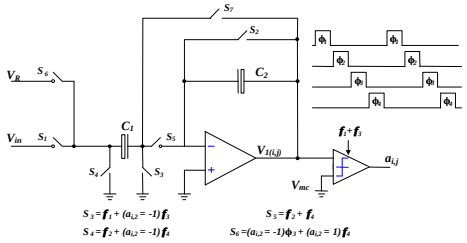

Figure 3. Circuit diagram of the analog part

Assuming ideal components, for the cycle number p, the integrator output voltage $\boldsymbol{V_1}$ is given by the equation:

$$(1)\begin{cases} V_{1(p,2)} = V_{1(p-1,4)} + V_{in} = pV_{in} - V_R \sum_{i=1}^{p-1} a_{i,3} \\ V_{1(p,4)} = V_{1(p,2)} - V_R a_{p,3} = pV_{in} - V_R \sum_{i=1}^{p} a_{i,3} \end{cases}$$

In addition, the integrator can be characterized by an input offset voltage $\boldsymbol{V_{off}}$ due to the charge injection of the switches and due to the amplifier offset. To compensate the effect of these, the conversion cycle used in ADMIR is divided in three periods, preceded by the reset of both the integrator and the up-down counter.

During the first period, $\boldsymbol{V_{in}}$ is converted in equation (1) as described previously, so the digital output code $\boldsymbol{N_1}$ corresponds to the analog value $\boldsymbol{V_{in}+V_{off}}$. The second period is used to invert the integrator output value $\boldsymbol{V_1}$, and the third period to convert $\boldsymbol{-V_{in}}$, so the digital output code $\boldsymbol{N_3}$ correspond to the analog value $\boldsymbol{-V_{in}+V_{off}}$. The difference $\boldsymbol{N}$ between $\boldsymbol{N_1}$ and $\boldsymbol{N_3}$ is equal to: $\boldsymbol{N = N_1-N_3} = 2\times\boldsymbol{V_{in}}$

More precisely, at the end of the first period (after m cycles), the integrator output voltage $\boldsymbol{V_1}$ is given by:

$$V_{1(m,4)} = \frac{C_1}{C_2}\left[mV_{in} - V_R \sum_{i=1}^{m} a_{i,3} + 2mV_{off} \right] \quad (2)$$

The second period is used to invert the integrator output voltage. The integrator output voltage becomes:

$$V_1 = -\frac{C_1}{C_2}\left[mV_{in} - V_R \sum_{i=1}^{m} a_{i,3} + (2m-1)V_{off} \right] (3)$$

At the end of the third period the integrator output voltage $\boldsymbol{V_1}$ is given by:

$$V_{1(2m,4)} = \frac{C_1}{C_2}\left[-2mV_{in} + V_R \sum_{i=1}^{m} a_{i,3} - V_R \sum_{i=m+1}^{2m} a_{i,3} + V_{off} \right] (4)$$

The dynamic range of $\boldsymbol{V_1}$ is given by $\boldsymbol{(\pm V_R+V_{off})C_1/C_2}$, so we have:

$$-\frac{C_1}{C_2}(V_R + V_{off}) \le V_{1(2m,4)} \le \frac{C_1}{C_2}(V_R + V_{off})$$

$$\Leftrightarrow -\frac{V_R}{2m} \le V_{in} - \frac{V_R}{2m}\left[\sum_{i=1}^{m} a_{i,3} - \sum_{i=m+1}^{2m} a_{i,3} \right] \le \frac{V_R}{2m}$$

$$\text{Then}: V_{LSB} = \frac{V_R}{m}N = \frac{1}{2}\left[\sum_{i=1}^{m} a_{i,3} - \sum_{i=m+1}^{2m} a_{i,3}\right]$$

Where N is the ADC output code and VLSB is the analog input voltage corresponding to the least significant bit. Thus, the conversion is independent of the offset voltage $\boldsymbol{V_{off}}$ as well as of the capacitor ratio $\boldsymbol{C_1/C_2}$.

An extra bit accuracy can be obtained by detecting the sign of $\boldsymbol{V_{1(2m,4)}}$ at the end of the conversion. The previous equations are replaced by:

$$-\frac{C_1}{C_2}\left(V_R + V_{off}\right) \le 2V_{1(2m,4)} - V_R a_{2m+1,1} \le \frac{C_1}{C_2}\left(V_R + V_{off}\right)$$

$$\Leftrightarrow -V_R \le 4mV_{in} - 2V_R \sum_{i=1}^{m}\left(a_{i,3} - a_{i+m,3}\right) - V_R a_{2m+1,1} \le V_R$$

$$\Leftrightarrow -\frac{V_R}{4m} \le V_{in} - \frac{V_R}{2m}\left(\sum_{i=1}^{p}\left(a_{i,3} - a_{i+m,3}\right) + \frac{1}{2}a_{2m+1,1}\right) \le \frac{V_R}{4m}$$

$$-\frac{C_1}{C_2}\left(V_R + V_{off}\right) \le V_{1(2m,4)} \le \frac{C_1}{C_2}\left(V_R + V_{off}\right)$$

$$\Leftrightarrow -\frac{V_R}{2m} \le V_{in} - \frac{V_R}{2m}\left[\sum_{i=1}^{m} a_{i,3} - \sum_{i=m+1}^{2m} a_{i,3}\right] \le \frac{V_R}{2m}$$

$$\text{Then}: \begin{cases} -\dfrac{V_{LSB}}{2} \le V_{in} - N \times V_{LSB} \le \dfrac{V_{LSB}}{2} \\ V_{LSB} = \dfrac{V_R}{2m} \\ N = \displaystyle\sum_{i=1}^{m}\left(a_{i,3} - a_{i+m,3}\right) + \frac{1}{2}a_{2m+1,1} \end{cases} \quad (5)$$

2.3. "Testboard" description

For testing the ADMIR prototype, a specific characterization board is needed. This characterization board must allow the validation of ADMIR test chips. The board must be able to be connected with a system realized with standard tools: analog generator, spectrum analyzer, clock generator, acquisition and generation board, it's the Dolphin Integration solution.

During the characterization phase, the following points have to be satisfied:

- Functional checking (conversion function, calibrations, standby...)
- Tests modes checking
- Power consumption measurement
- FFT curves
- SNR calculation
- THD measurements
- Common mode rejection measurements
- INL, DNL

For functional checking, a reconstruction DAC is used on the board. In the measurement phases, the analog inputs of the ADC must be precisely generated, and a digital code acquisition system is required (connection to an acquisition board). The principle is to apply for a pure sine signal on input of the ADC (by using a generator with a higher quality than the ADC to measure) to acquire the digital result and to analyze the digital results with software on a PC (Labview, FFT calculation...).

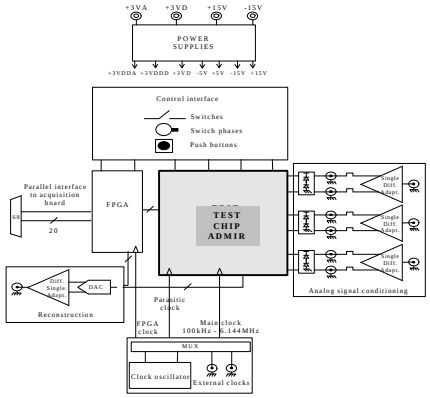

Figure 4. Characterization board block diagram

- Analog signal conditioning

The measurement systems present a single output analog interface available on a coax connector while the analog input of ADMIR can be either single-ended or differential. So we need to use single to differential adapter's devices whose main features must be better than the ADC features.

- Reconstruction DA converter

The reconstruction DAC is used to satisfy the functional checking. During this phase, this converter allows to connect the board with an analog measurement system.

- Power supplies

4 different voltages supply the characterization board:

- +3VA is the analog positive power supply of the test chip only,
- +3VD is the digital positive power supply of the test chip only,
- +15V is the positive power supply used in the analog conditioning device,
- -15V is the negative power supply used in the analog conditioning device.

The separation between +3VA and +3VD allows obtaining perfect isolation between analog and digital parts of the test circuit. With this configuration, the power consumption measurement can be achieved with independent variations on each part.

- Control interface

An FPGA is implemented in the "testboard". The purpose of this FPGA is twofold:

- to interface between the test chip and the acquisition board,

- to enable programmation of the test chip through switches on the board.

All those sub-blocks are modeled in VHDL-AMS to create the Virtual Test environment used for the analog testbench.

3. Mixed-Signal models using VHDL-AMS

For designing ADMIR and validating it, we have identified three different simulation models, with different purposes and level of refinements. Those models are used at different steps of the design methodology.

The first one is the "algorithmic level model", which describes the functionality and the key behavior, and which is used to specify the architecture. The second one is a "performance measurement model", and the third one is " block detail AMS model". The use of each model is described more precisely in section 3.

In this section some parts of the VHDL-AMS source code of the analog module, using discussed equations, are presented.

3.1. Algorithmic level model

This model is a mathematical description of the functionality of the Virtual Component. The purpose is to check that the expected specifications can be reached by the architecture. The analog part contains an ideal description of the architecture, and the digital part contains only the equations of the functions accomplished. For ADMIR, the digital part is an up-down counter that executes the integration operation needed in the digital calculation unit. Figure 5 shows the VHDL-AMS Algorithmic level model of the first order sigma-delta modulator using equation (1).

3.2. Performance measurement model

This model is used to verify "non-ideal behavior" of the analog part of the VC, the digital part is the same than in the "algorithmic level model". The purpose is to measure the influence of the variation of some parameters on the main characteristics of the component. In fact, different "performance measurement model's" can be developed, depending on the parameters that must be checked. In ADMIR, we have modeled the influence of variations of kT/C, of the gain of amplifiers, and of switch capacitor value, ...

The next model of the first order sigma-delta modulator with offset and charge injection conpensation uses equations (3, 4 and 5) of section 1. Generic parameters of this entity are used to perform different performance measurements of the model.

```
entity IncrementalADC is
port (quantity VR : in real; quantity VIN: out real; signal ai, phase : in integer);
end IncrementalADC;
architecture First of IncrementalADC is
begin
  if phase > 0 use
     VIN == VIN + VR*real(ai);
  else
      VIN == 0.0;
  end use;
end First;
```

Figure 5. Algorithmic model.

```
entity IncrementalADC is
generic (C1 : real := 1.0;  C2 : real := 1.0, VOFF : real := 0.0);
port (   quantity VR : in real;
         quantity VIN: out real;
         signal period, ai , reset: in integer);
end IncrementalADC;
architecture Second of IncrementalADC is
begin
 if reset = 0 use
    VIN == VIN;
 else
    case period use
     when 1 => -- first period equation
          VIN == (C1/C2)*(VIN - VR*real(ai)+ 2.0*VOFF);
     when 2 => -- second period; integrator output voltage
                    inversion
          VIN == ((C1/C2)*VOFF)- VIN;
     when 3 => -- third period
          VIN == VIN + (C1/C2)*(VIN - VR*real(ai)+
                    2.0*VOFF);
     when others => VIN == VIN;
   end case;
  end use;
end Second;
```

Figure 6. Non-ideal behaviors model.

3.3. Block detail AMS model

This model is pin-to-pin compatible with the effective VC, and can be used for the integration of the VC in the final chip. It describes the ideal behavior of the analog part, including the response to the digital control signals, the I/O impedance and the power consumption. The digital part is the RTL model. By using a behavioral level description, this model enables the VC integrator to do mixed-level/mixed-mode simulations at a faster rate than with the structural model.

The model of figure 7 shows a sample VHDL-AMS source that introduces controls of the analog module and the conversation phase and cycle. This model uses equations discussed in section 1.

```
architecture PinToPin of IncrementalADC is
begin
if (phi = -1) use
   VIN == 0.0;
   comp == 0.0;
   icomp == -1.0;
  else
    if (clock = 1) use
     VIN == VIN;
     comp == comp;
     if (VIN > 0.0) use
          icomp == 1.0;
     else
          icomp == -1.0;
    end use;
   else
    comp == (icomp+1.0) / 2.0;
    icomp == icomp;
    if ((integ1 = 0 ) and (integ2 = 0 ) and  (reverse = 0)) use
         VIN == VIN;
     else
         if (dif  > 0) use
                    if (integ1 = 1) use
                    case phase use
                      when 1 => VIN == VIN + (aip-ain);
                      when 3 => VIN == VIN - (icomp*(vrp-vrn));
                      when others => VIN == VIN;
                    end case;
         end use;
         if ((reverse > 0) and (phase = 3)) use
           VIN == -VIN;
         end use;
         if (integ2 = 1.0) use
                    case phase use
                      when 1 => VIN == VIN + (aip-ain);
                      when 3 => VIN == VIN - (icomp*(vrp-vrn));
                      when others => VIN == VIN;
                    end case;
         end use;
      else
          if (integ1 = 1) use
                    case phi use
                      when 1 => VIN == VIN + ((aip-ain)*2.0);
                      when 3 => VIN == VIN - (((icomp+1.0)/2.0*
                                          (vrp-vrn))*2.0);
                      when others => VIN == VIN;
                   end case;
          end use;
          if ((reverse > 0) and (phase = 3)) use
               VIN == -VIN;
          end use;
          if (integ2 = 1) use
                    case phase use
                      when 1 => VIN == VIN + ((ain-aip)*2.0);
                      when 3 => VIN == VIN - ((icomp-1.0+last)*
                                (vrp-vrn));
                      when others => VIN == VIN;
                    end case;
        end use;
       end use;
      end use;
   end use;
  end use;
end architecture PinToPin;
```

Figure 7. Pin-to-pin model

These VHDL-AMS examples show the power of this language, it makes it possible to write models in new ways. Rather than being restricted by a number of built-in primitives to build a more sophisticated model, the building blocks are now equations, mathematical functions and event-driven concepts. This leads immediately to the possibility of writing models using a behavioral approach where the model describes how a device behaves, not how it was implemented. The benefits of this approach are simplicity and better performance for simulation, without loss of accuracy.

4. Test script and environment

Virtual Test is intended to be used both by designer and Test Engineers, so it is important that the simulation and software models are as transparent as possible. The Virtual Test should start in a test script, and end with quantifiable test measurements back to the test script to make a pass/fail determination. This is achieved using the VHDL-AMS behavioral description of the device and the "testboard". Simulations are performed using SMASH, a mixed-signal mixed-level simulator. The test script is written in C++. The ActiveX API feature offered by SMASH, facilitate the simulation control and interactions between the simulator and the script program.

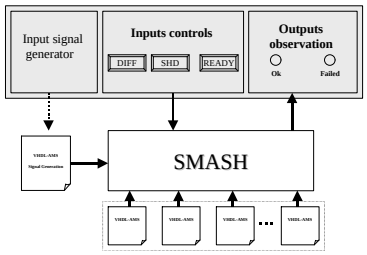

Figure 8. Test environment.

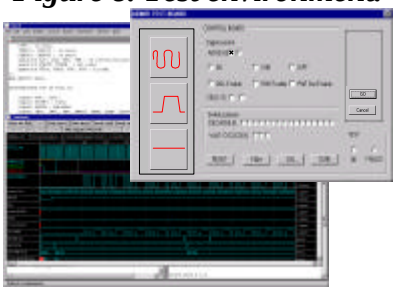

Figure 9. ADMIR Test with SMASH simulator.

The user interface (Figure 9) of the test environment is designed to reflect the "testboard". It has three modules:

- The "analog signal conditioning" module generates VHDL-AMS files modeling the input signal according to the user requirements. Figure 10 illustrates a generated file of a VHDL-AMS model of a sinusoidal signal.

- The “inputs control” module emulates all switches and push buttons of the “testboard”.

- The “output observation” module read simulation output and compares them to the reference simulation output.

```
entity Source is
    generic (    ampl:  real := 5.0 ;  freq: real := 100.0E+3 );
    port ( terminal P, M: electrical );
end Source;
architecture Sinusoid of Source is
    quantity VPM across  IPM  through P to M;
begin
    VPM == ampl * sin (2.0 * PI * freq * now);
end Sinusoid;
```

Figure 10. Input signal model.

5. Top down methodology

In the previous paragraphs, we have presented the architecture and the simulation models of ADMIR. Let us now present the design methodology used for ADMIR and see how and when the previously described models are used.

The virtual test equipment modeled in VHDL-AMS is a structure that enables to run the same simulation with different models, and it gives a reference for the good functionality of the VC. As a consequence, each model can be simulated and easily compared with another model to verify that the characteristics are similar and that no errors have been implemented in the new model. Furthermore, with the Virtual test equipment, we have developed a set of simulations which are considered as the “reference simulations”, that is to say simulations that are needed and sufficient for validating the ADC. All the reference simulations are applied to each model.

At the beginning of the design, simulations are run to verify that an incremental ADC can cover the working area required for the generator. The purpose is to specify the characteristics and the architecture of ADMIR, for addressing customer applications. The “algorithmic level model” is accurate enough for this purpose.

The second step of the design is to implement the architecture. It is then necessary to analyze the variations of parameters to see the influence on the global characteristics of the ADC. For example in ADMIR, the resolution should not change if capacitances were precise at ±30%. This goal is achieved thanks to the “performance measurements model”, which results are compared with the results of the “algorithmic level model”.

After that, the “block detail AMS model” is used for verifying the connection between the analog and the digital part, and the connection to the other blocks of the final chip. This model is accurate for testing the I/Os, for verifying the functionality, and it takes advantages of the behavioral modelization to run much faster than a simulation at a transistor level.

Finally, some simulations at the transistor level are needed, but it is not necessary to run such a simulation on the complete ADMIR Virtual Component.

6. Conclusions

This project demonstrates that with VHDL-AMS, a designer can implement a true topdown design methodology for analog and mixed-signal designs. Topdown design for digital systems has already shown that this improves productivity and concurrent engineering. The simulation output of the algorithmic level model and behavioral model of each subsystem, available at early stage of the design process, facilitate the analog testbench creation. This gives a debugging environment for the designer in the whole design process.

The creation of the test environment and subsystems models using a standardized language will be not viewed as an extra charge in the design process and will make models written in this language, tool and vendor independent and improve reusability of models.

Using ADMIR, the VC generator, in this experience is significant, we have to test and validate several cuts. Only this automated virtual test environment will permit a quick test of a high number of different cuts and configurations. This environment is important in VC concept and will be available for customer as ADMIR demonstrator.

This work has approached the virtual test and analog testbench using the ADMIR “testboard” example, this shows that virtual test concept can be integrated in the design process efficiently. Next step will be the model creation of standard test environment using VHDL-AMS.

References and Further Information

[1] Alain Vachoux, Analog and Mixed-Signal Hardware Description Languages, Kluwer Academic Publishers, 1997.

[2] A. Alegre de La Sougeole, Sigma-Delta Converters for Multimedia Applications, Analog&Mixed-Siganl Design Conference, July 1994, San Jose.

[3] Dolphin SMASH user manual.

[4] Jean-Michel Berge, Modeling in Analog Design, Kluwer Academic Publishers, 1995.

[5] Jacues Robert, A 16-bit Low-Voltag CMOS A/D Converter, IEEE Journal of Solid-State circuits, Vol. 22, April 1987.

[6] Jacues Robert, A Second-Order High-Resolution Incermental A/D Converter with Offset and Charge Injection Conpensation, IEEE Journal of Solid-State circuits, Vol. 23, June 1988.

[7] VHDL-AMS (IEEE 1076.1) Standardisation committee http://www.vhdl.org/analog