computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

INDUSTRIAL TEST OF A/D CONVERTERS IN LABVIEW

Stéphane Raze [1,2], Dominique Dallet [2] , Phillipe Marchegay [2]

1) Serma Technologies, 30 av. G. Eiffel, F-33608 PESSAC CEDEX, FRANCE
e-mail : s.raze@serma.com, url : http://www.serma.com
2) Laboratoire IXL, Université Bordeaux I, 351 cours de la Libération, F-33405 Talence CEDEX, FRANCE
e-mail : dallet@ixl.u-bordeaux.fr, url : http://www.ixl.u-bordeaux.fr

Abstract: *An industrial test bench for Analog-to-Digital Converters (ADC) is presented in this article. The bench hardware has been composed with known devices of instrumentation. This hardware had been identified from ADCs under test requirements. All of hardware is handled by a software written in LabVIEW development system.*

Keywords: *ADC dynamic test, LabVIEW*

1. INTRODUCTION

An ADC is a mixed device which converts an analog voltage to a numerical output on n bits (n is called resolution).
ADC transfer function is defined as [1]:

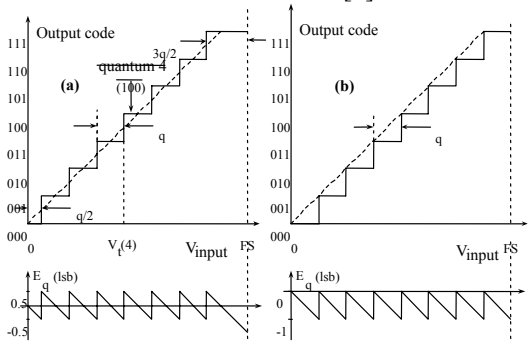

Fig. 1 - Transfer/error function of a theoretical 3 bits ADC by rounding (a) and truncation (b).

The general principle of ADC characterization is as following :

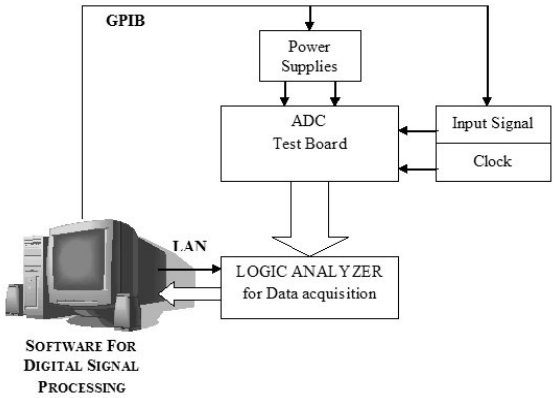

Fig. 2 - ADC Test Scheme.

2. ADC TESTING METHODS

ADC dynamic test methods are well known [2] and some standards [3][4][5] have been presented. Those numerical analyses are essentially from three types :

- Time analysis : interpolation allows getting the best fit of output reconstructed sinusoid. A comparison with theoretical input signal allows noise estimation and so SNR calculation.
- Spectral analysis : a Fast Fourier Transform (FFT) is used to calculate parameters like SNR, SINAD, THD, SFDR, and ENOB (effective number of bits),
- Statistical analysis : a histogram shows how many times any different code appears on ADC output. It is compared to theoretical and then linearity, offset and gain errors can be calculated.

Testing a component in an industrial context is something different from testing in a lab. In fact, when a customer asks to Serma Technologies for testing a component, the reference document is the manufacturer's datasheet. So, for functional tests of ADCs, only spectral and histogram analyses are used. That's why time fitting analysis is not presented in this article.

3. TEST BENCH HARDWARE : A TOTAL MODULARITY.

SIGNAL GENERATION

Choice of signal generators is made easy by the great quantity of models that are available on the instrumentation market. Clock generator must have as low jitter as possible. For the input signal

generator, most important parameters are noise level, which must be lower than ADC's noise, harmonic distortion (for the synthesizers) and phase dispersion.

Thus, modularity for the part "signal generator" of the test bench is simply to choose a signal generator which complies with ADC under test requirements. This flexibility is the must important fact that justify using a modular test bench and not an ATE system.

The generators owned by Serma Technologies are presented in Table 1 :

Table 1. Models of available signal generators

Generator	Freq. Range	ADC resolution
HP8656B	0.1 – 990 MHz	< 10 bits
HP8648B	0.1 – 2000 MHz	< 10 bits
HP8904A	DC – 600 kHz	< 16 bits
HP33120A	DC – 15 MHz	< 12 bits

Passive filtering is used on input signal to limit harmonic distortion of generators.

For ADCs with mid-resolution/mid-speed (typically 14-bits/5 MHz), some generators are available (HP866x, Rhode&Schwarz SMLxx) but unaffordable to buy. In this case, a company like Serma Technologies could rent such a generator.

In addition, we own a Lecroy 9210 pulse generator (up to 250 MHz) for ADCs that need square clock signal.

4. COLLECTING THE ADC OUTPUT DATA

The easier way to do that is to use a logic analyzer. As for signal generators, acquisition requirements (that are length i.e. memory, speed, threshold voltage) are a function of the ADC under test requirements.

That's why we have chosen a mainframe with modules (data acquisition cards). The Agilent HP16700 family look like a UNIX (hp-ux) based mainframe with five extension slots for use data acquisition, pattern generation and digital oscilloscope cards.

We have equipped it with two data acquisition boards :

Board	*Pods*	*Memory*	*Speed*
HP16517A	2 x 8 bits	64k words	0-333 MHz
HP16717A	4 x 16 bits	2M words	20-1000 MHz

Considering that reasonable minimum number of samples for testing an *n* bits ADC is :

- $N_1 = \pi \cdot 2^n$ for FFT analysis,
- $N_2 = 16 \cdot 2^n = 2^{n+4bits}$ for histogram analysis,

our system allow us to test any converter up to 16 bits of resolution (up to 20 bits only for FFT analysis).

5. LabVIEW IMPLEMENTATION

Programming signal generators

LabVIEW drivers for generators are available on either LabVIEW CDROM, or National Instruments website. But we must choose a generator without changing the code of the program. Thus, we developed (as shown in Figure 2) a VI that handles any generator using its own GPIB words. In fact we just need words for amplitude level, frequency, units, and eventually an initialization string.

So, the test program allows the user to choose a generator in a menu. The menu is build dynamically at the program start by reading a text file named "listgen.dat" which contains the available generators list. Each item is in fact the name of a file which contains the remote commands of the generator in a LabVIEW cluster format.

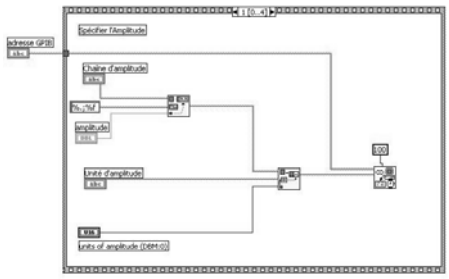

Fig. 3 - Remote handling of a generator.

In addition, we have developed a VI for creating, saving and loading a cluster "Generator".

Programming the logic analyzer

Configuration of the logic analyzer for getting the data of the ADCs under test is relatively simple :

- choose of the acquisition card,
- setting of the bits to be acquired
- setting of the voltage threshold
- setting the format of data (bin/hex/dec)
- finally save the configuration in a file

The HP16700 logic analyzer is handled by the LAN (Local Area Network) with a library of Unix-like commands developed by Agilent [6] : the RPI (Remote Programming Interface). Using TCP-IP protocol on port 6500 of the HP16700, the fundamental use of the logic analyzer for our application is :

- load the previously saved configuration file
- run the acquisition
- download the acquisition data on the PC (within the LabVIEW program).

The part of code in fig. 4 does the 3 operations

above :

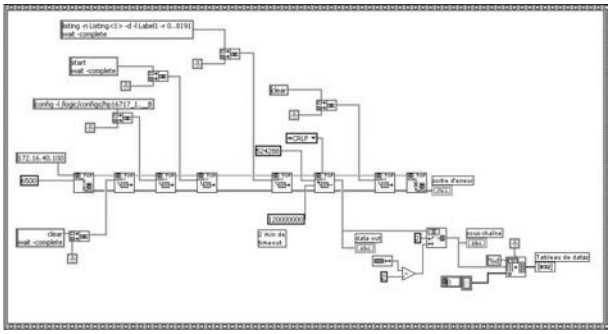

Fig. 4 - Remote handling of the logic analyzer.

Digital signal processing

The use of LabVIEW is generally associated with calculation and signal processing. LabVIEW has got lots of VIs in this domain [7].

For example, some VIs for FFT computation have been used as they are in our software. Besides, some changes could be necessary for SINAD or THD calculation when the input frequency goes up to sample frequency (aliasing of harmonics). In addition, the histogram and comparison of theoretical and experimental density of probability for example is specific of ADC testing. That's why we developed such VIs.

Spectral Analysis

First of all, coherence of frequencies is available in our application. The user may choose between no-coherence, signal-coherence or clock-coherence, and a Vi named "*coherence.*vi"calculates the modified frequencies. The VIs used for single-tone spectral analysis are shown in Fig. 5 below. LabVIEW intrinsic VI named "AutoPower spectrum" has been used in our program but VIs for calculation of parameters *SINAD*, *THD*, *ENOB, SFDR*, and *SNR* were not usable (buggy or too much complex).

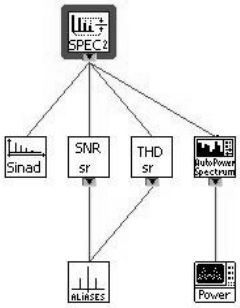

Fig. 5 - Single-tone analysis in LabVIEW.

Similar VIs have been created for dual-tone spectral analysis. A sub-VI named "*aliases.vi*" has been written to calculated aliased bins of harmonics (for those that exceed Nyquist frequency : half of sampling frequency).

Apodization windows (Rectangular, Hamming, Hanning, Blackman, Blackman-Harris 4 & 7, Kaiser) are available for use in our

So we have created those custom VI whose are optimized for our application.

Statistical Analysis

Like for spectral analysis, some VIs are available directly in LabVIEW for histogram calculation. But they are very complex. Our use of histogram is simply counting the number of occurrences of each code in an acquisition.

The Fig. 6 below shows our optimized LabVIEW implementation of the histogram algorithm.

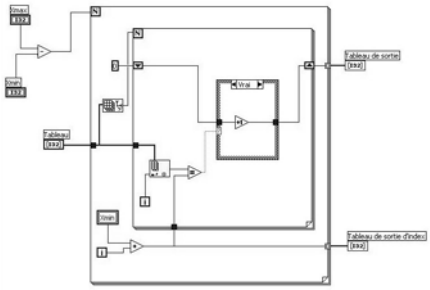

Fig. 6 - histogram count in LabVIEW.

Fig. 7 shows the VIs used for *DNL* and *INL*, *Missing codes*, *Gain* and *Offset* calculation.

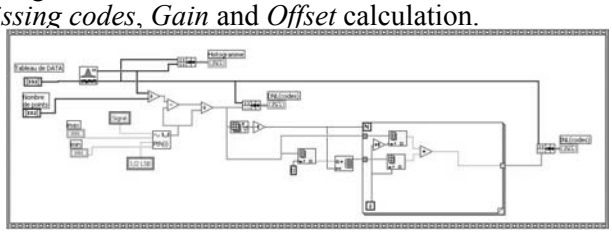

Fig. 7a - DNL & INL in LabVIEW.

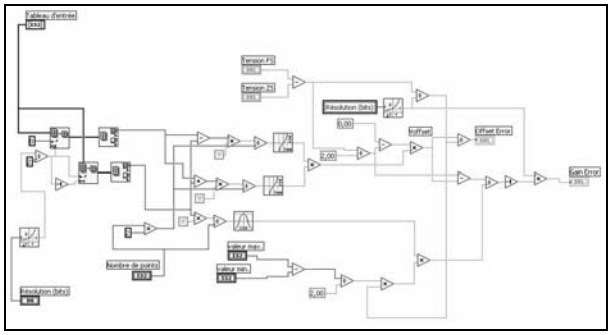

Fig. 7b - Gain & Offset in LabVIEW.

Complete program

Graphical user interface (GUI) of the complete test program is shown in Fig. 8.

On the left, some tabs allow the user to specify the ADC under test, the parameters of the acquisition (amplitude levels, frequencies, number of samples, type of generators), the setup of logic analyzer, and some options such as FFT windowing, serial/parallel data, analysis (spectral/statistical), direct of multiplexed output, file selection for data log of the results, file selection for saving the acquisition data …

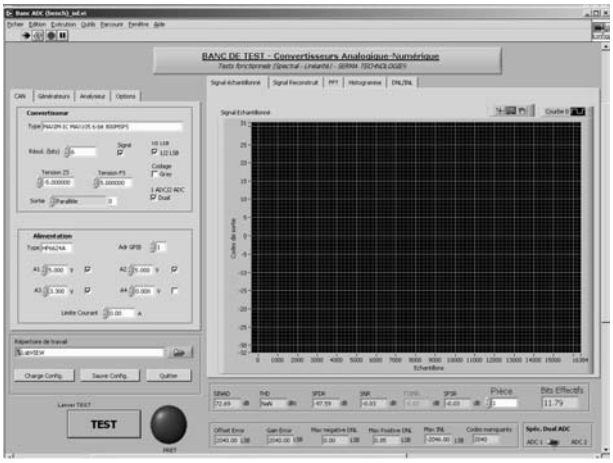

Fig. 8 - GUI of the bench test program.

Under those tabs, three buttons allow to save, load a configuration file or stop the program. When saving a setup, all elements in one tab are packed in a LabVIEW cluster and all four clusters are packed again in a big cluster which is the format of the configuration file.

This big cluster is copy-pasted in a global variable, so when reading a configuration file, the program is able to "un-clusterize" the previously saved data.

The larger part of the interface is graphical results (in tabs) for sampled/reconstructed signal, FFT spectrum, histogram, differential and integral non-linearity, and transfer characteristic.

Finally, at the bottom of screen, parameters measured are presented in numerical format.

Test bench validation

In order to verified the algorithms applied in our program, a 50 MSPS 10-bits Flash ADC (Texas Instruments THS1060) had been tested.

The test setup was as following :

- Input signal at 5 MHz with Agilent E4400 generator and passive low-pass filter
- Clock signal at 50 MHz with HP8656B generator
- 65536 samples

Spectral analysis results are presented in Table 3:

Table 3. Test bench results for FFT analysis

SINAD	THD	SNR	SFDR	ENOB
57.39 dB	-72.59 dB	57.47 dB	73.21 dB	9.25 bits

This results are the same than those delivered by CanTest system [9].

From bench test to production test

Beyond the scientific characterization of components, a company like Serma Technologies needs to test components in production flow, with quantities defined by the customers. That's why we developed a "production mode" version of our test software. The GUI for this program is presented in Fig. 9.

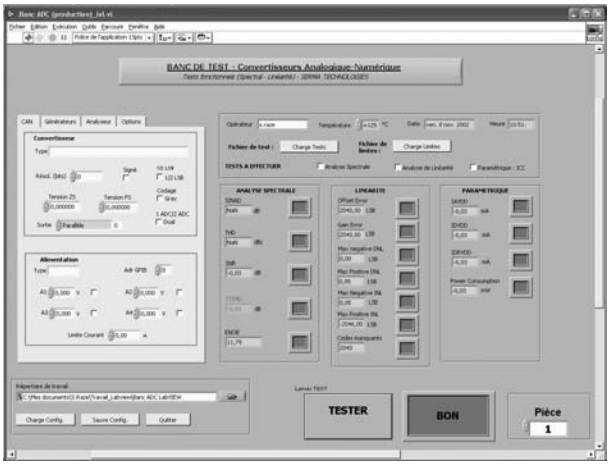

Fig. 9 - GUI of production test program.

In this program, graphs have been replaced by Booleans. In fact, the left part of GUI is unchanged. But the right part is something different. Top of the right part contains information for data log that are : the name of the operator, the date & time of test, temperature of test, a button to load a setup, and another to load a file containing limits for the parameters to be measured.

Those limits are specified in the manufacturer's data sheet.

For each parameter, if its value is between the limits, the Boolean is true, otherwise it's false.

After the test is complete, a "AND" operation is realized on all Booleans and a big Boolean is resulting (Good/Bad component). At this time the number of tested pieces increments and the data log file is updated with all the values : name of component, name of operator, temperature, date, time, and for all parameters measured, value, limits, and text : PASS or FAIL.

6. CONCLUSION

A LabVIEW based software for ADC test has been presented in this article. Its advantage is its modularity, from software and hardware. Speed problems from LabVIEW code execution have been detected and those parts will be replaced by parts written in C language. At the moment, this document is not innovating, other software [8][9] being existing. Moreover, parametric tests will be added, to measure voltages, currents, et resistances at component pins.

From the same principle, a study on a DAC test bench will follow, and another concerning precision operational amplifiers.

7. REFERENCES

[1] D.H. Sheingold. *Analog-Digital Conversion Handbook, 3rd edition, Analog Devices, ed*. Prentice-Hall, Englewood Cliffs.

[2] J. Doernberg, H.-S. Lee and D.A Hodges. *Full-Speed Testing of A/D Converters*, IEEE J. Solid-State Circuits, December 1984, SC-19, pp.820-827.

[3] IEEE Draft Standard 1241, "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters", February 2000.

[4] EUPAS : European Project for ADC-based Devices Standardisation.

[5] C. Morandi et al. "DYNAD : a framework IV SMT project addressed to the development of dynamic test techniques for analog-to-digital converters", Computer Standard & Interfaces, vol. 22,no. 2, pp113-119, June 2000.

[6] Remote Programming Interface (RPI) for the HP 16700 Logic Analysis System, © Copyright Hewlett-Packard Company 1999

[7] LabVIEW Analysis VI Reference Manual

[8] "ADC Test Data Evaluation Program for Matlab" Home Page, http://www.mit.bme.hu/services/ieee/ ADC-test.

[9] S. RENAUD – Contribution à la caractérisation des circuits de conversion analogique numérique. Conception et réalisation d'un système d'évaluation dynamique de ces dispositifs. Thèse de Doctorat, Université Bordeaux I, 1990.