IMTC 2007 – IEEE Instrumentation and Measurement
Technology Conference
Warsaw, Poland, May 1–3, 2007

A Generic ASIC Architecture for Real Time Time-Frequency Analysis of Non-stationary Large Bandwidth Signals

L. Noury[1], F. Durbin[2], H. Mehrez[1], A. Tissot[2]

[1]Laboratoire d'informatique de Paris 6,
Université Pierre et Marie Curie
4, place Jussieu – 75005 Paris, France
E–mail: {ludovic.noury, habib.mehrez}@lip6.fr.

[2]DAM/DIF/EIM
Commissariat à l'énergie atomique
91680 Bruyères-le-Chatel, France
E–mail: {francois.durbin, andre.tissot}@cea.fr.

Abstract – *Wide-band time varying digital signals are extensively used by GSM, TV relays or radars. Analyzing those signals requires building a Time Frequency Representation (TFR) in real-time. Most of TFRs are derived from the Wigner-Ville TFR; unfortunately they are computationally too intensive for real time processing. In this paper we propose an alternative approach to design an ASIC which computes a particular TFR of an input signal in real-time that we named F-TFR. Our approach is based on gradual signal channelizing using modulation, FIR filtering and time-interleaving of the channelized signals. After a brief introduction, we describe in detail the F-TFR algorithm. Then, we study the corresponding hardware architecture introducing signal interleaving and equations adaptations which are necessary for hardware implementation. Finally, we present some typical signals analysis by F-TFR along with an hardware prototype.*

Keywords – *time-frequency analysis, Wigner distributions, FIR digital filters, very-large-scale integration, frequency-channelizing.*

I. INTRODUCTION

Nowadays, high-frequency digital signals are commonly used in many telecommunication applications such as GSM, radio, TV relays or radars. Unfortunately these signals are particularly difficult to analyze.

First, they contain multiple time-varying frequencies which cannot be monitored with classical discrete Fourier transform (DFT) based analyzers since DFT, defined by :

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N} \tag{1}$$

which implies that signal frequencies are stationary. Hence, there is no influence of time in the result and fast varying frequencies cannot be monitored adequately.

We take for example the signal $x_{chirps}(n)$ composed of two linear chirps, the first with a linearly increasing frequency and the second with a linearly decreasing frequency:

$$x_{chirps}(n) = \cos\left(2\pi(f_0 nt_s + \frac{k}{2}(nt_s)^2)\right) + \cos\left(2\pi(f_1 nt_s - \frac{k}{2}(nt_s)^2)\right) \tag{2}$$

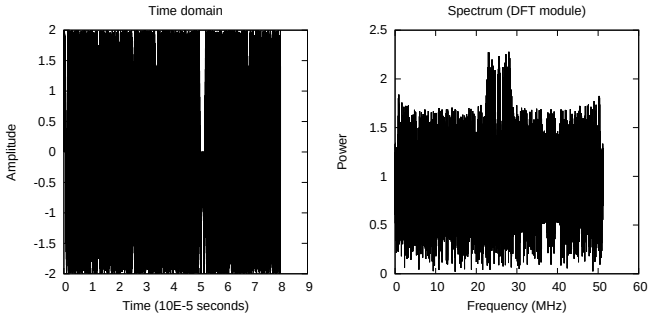

Fig. 1. Left: time domain – $x_{chirps}(n)$. Right: 8196 points DFT spectrum – $X_{chirp}(M)$. Neither the time representation nor the frequency analysis shows the signal characteristics.

The corresponding DFT analysis figure 1 does not exhibit the signal properties, with its two frequencies varying linearly with time. An analysis involving both time and frequency is necessary to display these informations. Such an analysis is called a Time-Frequency Representation (TFR) of the signal.

Moreover, these signals often have a wide spectral band of interest ($>$ 50Mhz) that must be monitored in real time. Most TFRs are derived from the Wigner-Ville TFR [1] defined by:

$$WV(f,t) = \int_{-\infty}^{\infty} x(t+\frac{\tau}{2}).x^*(t-\frac{\tau}{2}) e^{-j2\pi f\tau} d\tau \tag{3}$$

which requires a high computational capacity as it needs $N \times 2FFT_N$, with N the number of discretization points in time and frequency, leading to a complexity of $O(N^2 log_2(N))$.

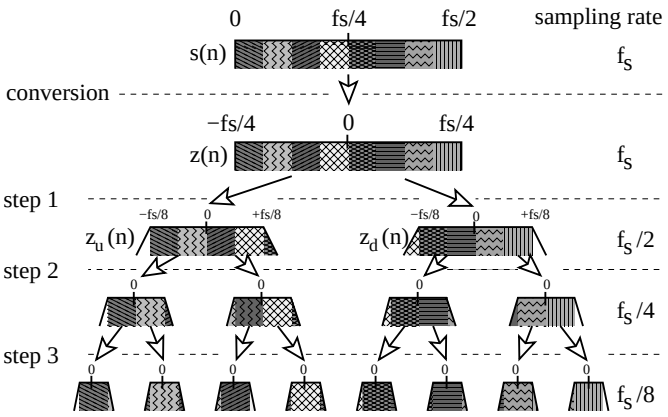

Fig. 2. Frequency channelizing algorithm illustration

As a consequence, analyzers performing real-time Wigner-Ville based time-frequency analysis of wide-band signals cannot be easily designed. [2] describes a Wigner-Ville based analyzer prototype using 9 units of TI TMS320 series DSP dedicated to 3G Telecommunication signal but which processes only one block of $N = 512$ signal samples.

Moreover, since we aim at reducing the TFR analyzer size by designing a suitable architecture for an embedded system, we have the VLSI (Very Large Scale Integration) constraints of an ASIC (Application Specific Integrated Circuit) or of a SOC (System On Chip). These constraints (area, frequency and consumption limitation) and the $O(N^2 log(N))$ Wigner-Ville TFR complexity explain why, as far as we know, there is no hardware implementation of a Wigner-Ville TFR today.

But [3] demonstrates that another equipment known as RFEL PFT [4] which is a FPGA multi-channel filter-bank, can be used as a time-frequency analyzer by proving that PFT results are similar to those given by a Wigner-Ville TFR.

This situation leads us to design a new ASIC architecture, named F-TFR (Fast Time-Frequency Representation) allowing real-time time-frequency analysis of wide band signals. This architecture, like PFT, is based on time-domain filtering to realize a time-frequency analysis. We insist on the fundamental fact that, while all computations are made in the time domain, results belong to the time-frequency domain.

II. THE F-TFR ALGORITHM

The input signal is gradually channelized into N signals, where N is the number of desired points in the frequency domain (illustrated figure 2). Each output signal amplitude corresponds to the input signal spectral contents in a fixed frequency-band.

A. Real input signal conversion to complex signal

The system input is a real digital signal $s(n) = x_n$ centered on $IF = f_s/4$ (IF: Intermediate Frequency) with a sample-rate $f_s = 1/t_s$, the band of interest $bp = [0; f_s/2]$ corresponds to the maximum usable band.

$s(n)$ is converted into a complex signal $z(n)$ centered on $IF = 0$, using modulation by a complex signal of $-f_s/4$ frequency. Hence, $z(n)$ is defined by:

$$z(n) = s(n).e^{-j2\pi\frac{f_s}{4}nt_s} \qquad \text{with } j^2 = -1 \tag{4}$$

Replacing f_s by $1/t_s$ gives:

$$z(n) = x_n.e^{-j\frac{\pi}{2}n} \tag{5}$$

$z(n)$ has a sample rate $f_z = f_s$, the band of interest $bp = [-f_s/4; +f_s/4]$ uses only half of the available band. We keep $z(n)$ oversampled by 2 to reduce filter constraints and aliasing.

B. Frequency channelizing

In a first step, we create two new signals $z_{up}(n)$ and $z_{down}(n)$ from $z(n)$. The frequency band $b_{up} = [-f_s/8; +f_s/8]$ of $z_{up}(n)$ corresponds to the $[-f_s/4; 0]$ $z(n)$ frequency band. Also the z_{down} band $b_{down} = [-f_s/8; +f_s/8]$ corresponds to $z(n)$ $[0; f_s/4]$ band. These 2 signals common sample rate is $f_1 = f_s/2$.

In a second step, the operation is repeated on z_{up} and z_{down} creating 4 new signals z_{uu}, z_{ud}, z_{du} and z_{dd} with a sampling rate $f_2 = f_s/4$. Each of these signal has a band of interest of $[-f_s/16; +f_s/16]$ corresponding to the $z(n)$ $[0; f_s/8]$, $[f_s/8; f_s/4]$, $[f_s/4; 3f_s/8]$ and $[3f_s/8; f_s/2]$ frequency bands.

This channelizing operation is repeated on all signals created at the previous step, until obtaining N signals after $S = \log_2(N)$ steps (see figure 2).

C. Basic channelizing operation

Each basic channelizing operation takes a complex input signal $z_x(n)$ centered on $IF = 0$ (A on figure 3). The band of interest is $[-f_x/4; +f_x/4]$ with a sample rate $f_x = f_s/c$; c is the current step number (count begins at 1). The input signal is oversampled by two.

First, $z_x(n)$ is modulated twice constructing 2 new signals:

$$z_{xu}(n) = z_x(n).e^{+j2\pi\frac{f_x}{8}nt_x} \tag{6}$$

and:

$$z_{xd}(n) = z_x(n).e^{-j2\pi\frac{f_x}{8}nt_x} \tag{7}$$

$z_{xu}(n)$ contains the $z_x(n)$ signal low frequencies centered on 0 and $z_{xd}(n)$ contains the $z_x(n)$ signal high frequencies centered

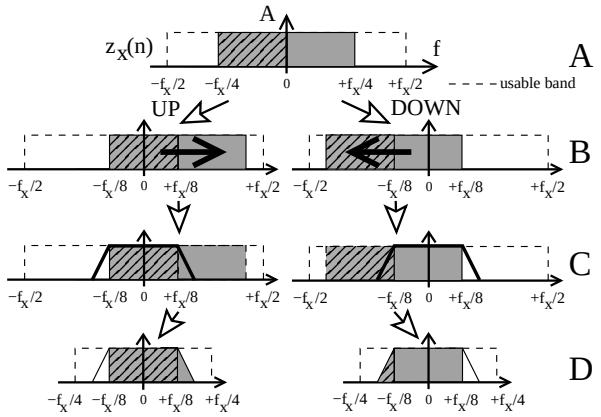

Fig. 3. Basic channelizing operation

on 0 (B on figure 3). Equations 6 and 7 can be simplified by replacing f_x by $1/t_x$:

$$z_{xu}(n) = z_x(n).e^{+jn\frac{\pi}{4}} \tag{8}$$

$$z_{xd}(n) = z_x(n).e^{-jn\frac{\pi}{4}} \tag{9}$$

Then, these 2 complex signals are filtered using 4 low-pass FIR filters, keeping only the frequency band $[-f_x/8;+f_x/8]$ (C on figure 3). The transition band is outside this interval in order to keep all the frequency contents, but at the cost of possible signal duplication on two neighboring frequency channels (also called frequency bins in the sequel, referring to the DFT bins).

Finally, since the two new signals band of interest is $[-f_x/8;+f_x/8]$, corresponding to $1/4$ of the usable band, we can decimate by 2, keeping an oversampled by two signal (D on figure 3).

III. HARDWARE ARCHITECTURE

The software to hardware link is particularly tight for F-TFR since each algorithmic step yields exactly one hardware stage. Hence, $N = 1024$ corresponds to $S = 10$ algorithmic steps or hardware stages.

A direct hardware implementation would use about $(N-1)/2$ basic channelizing blocks (BCB) or 1023 BCB for $N = 1024$ and $S = 10$, leading to a practically unfeasible ASIC. Fortunately, with a closer look at the signals sampling rate variation (figure 2) we realize that the computing hardware used at each stage can always be reduced to a single interleaved basic channelizing block (IBCB) giving 10 IBCB for $N = 1024$ (10 stages).

At each stage, the number of signals is doubled, but their sampling rate is divided by two. Hence, while at the first stage there is one signal sampled at f_s, there are two signals at $f_s/2$ at the second stage, then 4 signals at $f_s/4$ or, generally 2^{c-1} signals with a sampling rate of $f_s/2^{c-1}$ at the c^{th} stage.

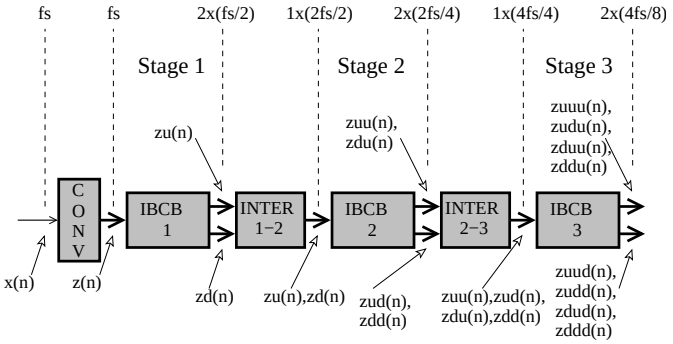

Fig. 4. 3 stages F-TFR block diagram

Using time multiplexing, these signals can be interleaved into one signal at f_s. This allows to use only one basic channelizing block per stage. The corresponding new hardware includes of a converter and a succession of N IBCB with an interleaving block between two IBCBs (see figure 4).

A. The $s(n)$ to $z(n)$ converter

By using Euler's identity, equation 5 becomes :

$$z(n) = s(n)\cos\left(n\frac{\pi}{2}\right) - js(n)\sin\left(n\frac{\pi}{2}\right) \tag{10}$$

The geometric representation of $z(n)$ on the unit circle shows that the only possible values for $\sin(n\pi/2)$ and $\cos(n\pi/2)$ are $-1;0;+1$, corresponding to the 4 points on the circle at angles $(0,\pi/2,\pi,3\pi/2)$.

This leads to an hardware implementation consisting of a 4-states finite state machine (FSM) synthesizing $z(n) = I_n + jQ_n$ by selecting values for the I and Q outputs from the set $\{-s(n);0;+s(n)\}$.

B. IBCB up and down modulators

Similarly, equation 8 yields:

$$z_{xu}(n) = I_n\cos\left(n\frac{\pi}{4}\right) - Q_n\sin\left(n\frac{\pi}{4}\right) + j\left[Q_n\cos\left(n\frac{\pi}{4}\right) + I_n\sin\left(n\frac{\pi}{4}\right)\right] \tag{11}$$

This time, the unit circle is divided into 8 equal arcs and the corresponding sine and cosine values are in the set $\{-\sqrt{2}/2;-1;0;+1;+\sqrt{2}/2\}$. We use the same argument for the down modulation equation 9.

Thus, the hardware consists of an 8-states FSM computing the outputs Iu, Qu, Id and Qd (with $z_u = Iu + jQu$ and $z_d = Id + jQd$) by selecting values among the set $\{\pm I_n, \pm Q_n, 0, \pm\sqrt{2}/2I_n, \pm\sqrt{2}/2Q_n\}$. For multiplication by $\sqrt{2}/2$ we use two canonical recoding constant multipliers [5].

But as we interleave the signals to keep one IBCB block per stage, the modulator input z_c contains $i_c = 2^{c-1}$ interleaved

signals. Consequently, the FSM must wait i_c cycles between two state switchings.

C. IBCB Filters

The 4 FIR filters used at each stage are the core of this architecture, they are on the critical path and thus determine the maximum operating frequency. Moreover, since those filters use a significant proportion of the ASIC area, they must be as small and fast as possible. All the four filters use the same M coefficients.

The selected filter architecture is based on our previous work [6] and again uses Dadda summation [7] and Fadavi-Ardekani algorithm [8] to avoid internal sign extension. The multipliers are now canonical recoding constant multipliers, as those used for the modulator. Also, the coefficients generation is constrained to return symmetrical coefficients. This allows a factorization which halves the number of multipliers.

In order to work on interleaved signals, these filters must have additional delays. The filtering operation :

$$y(n) = \sum_{i=0}^{M-1} a_i x(n-i-1) \tag{12}$$

is a convolution product which needs the M previous signal samples values to compute a result. Thus while the interleaving structure avoids to duplicate the computation hardware at each stage, the number of signals to be filtered doubles accordingly. Therefore, we must double the number of memorizing elements (registers, RAM) at each stage to follow this growth.

This imposes a stringent constraint on the FIR filter length: using more coefficients reduces signal overlapping, but increases latency and chip area.

Also, with the 2-decimation, the filter output is not valid at each cycle. At the first stage it is valid 1 cycle out of 2, on the second stage 2 cycles out of 4 and so on because each interleaved signal is decimated by 2. Hence, at each architecture stage c the filtering operation is applied to a sample of each of the $i_c = 2^{c-1}$ interleaved signals during i_c clock cycles, then during the next i_c clock cycles no filtering operation occurs.

D. Interleaver

The interleaver multiplexes in time the 2 complex output signals of the IBCB into a single complex signal. This signal is then used as an input for the next stage IBCB (figure 4).

The interleaver also synchronizes two consecutive IBCBs. The first IBCB, due to the 2-decimation, produces 2 complex samples during i_c clock-cycles, then no results occur during the next i_c clock-cycles and so on. On the other hand, the second IBCB consumes one complex sample per clock cycle. Hence, the interleaver must have some memorizing elements to act also as a synchronizing FIFO between two consecutive stages.

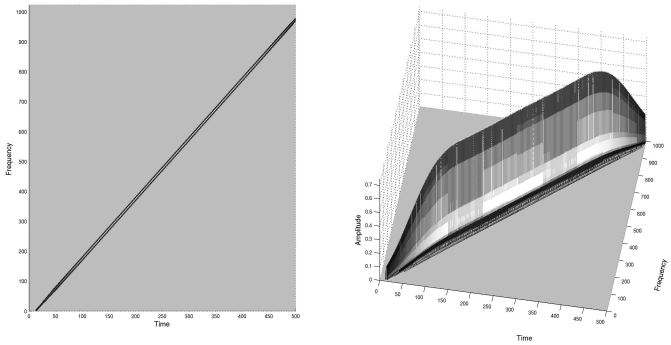

Fig. 5. F-TFR analysis of a linear chirp.

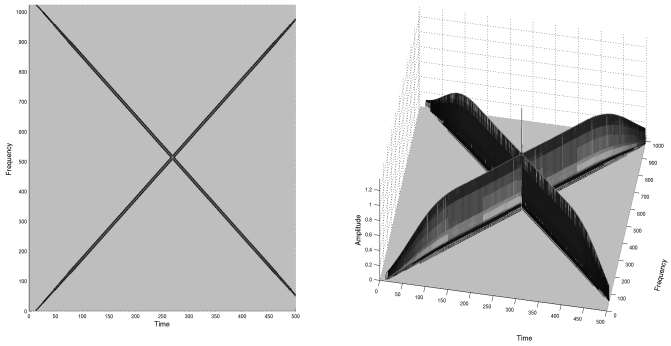

Fig. 6. F-TFR analysis of $x_{chirp}(n)$, the sum of two linear chirps.

IV. RESULTS

As far as we know, there is currently no other ASIC dedicated to time-frequency analysis, despite the growing need for large bandwidth non stationary signal analysis. As mentioned above, it is because implementing a Wigner-Ville derived architecture would need too large an area and could hardly meet high-frequency constraints.

By using a modulation/filtering/decimation/interleaving approach we have defined a generic ASIC architecture, suitable for large bandwidth non stationary signal time-frequency analysis, performed in real time.

A. F-TFR analysis results

Results from a 10 stages F-TFR ($N = 1024$) using 20 coefficients FIR filters are shown figure 5, 6 and 7.

The signal $x_{chirps}(n)$ (defined by equation 2) Fourier analysis using a 8196 dots DFT was inconclusive (figure 1). But, by using F-TFR, the Time-Frequency analysis on figure 6 clearly shows the linear frequency variation of each chirp component and the corresponding spectrum amplitude, with only $N = 1024$ bins. The F-TFR analysis also explains the DFT amplitude peak figure 1; figure 6 shows clearly that it corresponds to the only common point in the time-frequency domain of the 2 chirps components.

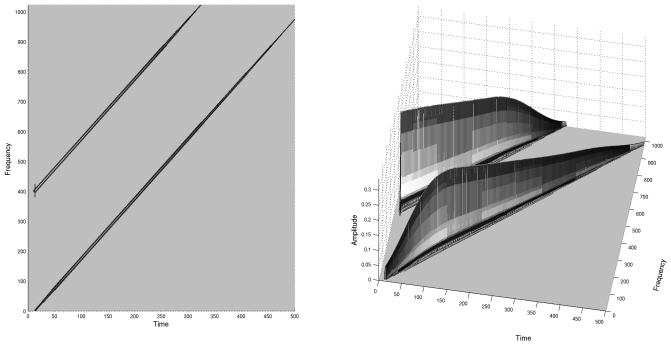

Fig. 7. F-TFR analysis of two Gaussian chirps

For each result (figures 5, 6 and 7), whereas the analyzed signal begins at $t = 0$ the analysis result begin at $t_r > 0$. This latency between the input signal and the analysis results is due to the multistage FIR filter and 2-decimation. It depends on the total number of stages S and on the FIR coefficients number M. By extensive test, we found the latency value expressed in clock cycles :

$$latency = 2^S.\frac{M}{2} \tag{13}$$

This may seem a significant value, but we must remind that the output frequency $f_S = f_s/2^S$ is much lower that the input signal frequency while the chip single clock rate is f_s.

We know that the theoretical Wigner-Ville TFR, for chirps, is constant in amplitude. We observe here that the F-TFR amplitude decreases at the lower and higher frequency spectrum ends while it is correct for the rest of the frequency bandwidth. This phenomenon is caused by the first stage FIR filter transition band. The FIR filter transition band decreases when we increase the number of filter coefficients M while the latency increases with the number of coefficients. A trade-off must be made, depending on the specific application in view.

B. Hardware

All these results were obtained by bit true simulation of the F-TFR architecture. The final validation consists in the manufacturing of a F-TFR ASIC prototype. This prototype is targeted to the architecture validation and testability. A high operating frequency is not the main goal, as the area must be minimized in order to limit the manufacturing cost.

A parameterizable F-TFR netlist generator was developed using *Alliance* [9] and *Stratus* [10] [11] CAD open-source CAD tools. The ASIC prototype uses ST microelectronics HCMOS9GP standard cells library.

The chip back-end was done with Cadence using a single clock implemented by a clock tree and a synchronous reset. The design is pad limited.

Finally, this prototype was manufactured using ST Microelectronics 0.12μ process, the main characteristics (surface, frequency, ...) are given on table I.

Surface	$3.39mm^2$ (core : $1.47mm^2$)
Technology	0.12μ (ST, HCMOS9GP)
Pads	80 (56 I/O, 24 VDD/VSS)
Transistors	848351 (418548 for memory)
Critical path	$6.632ns$
Estimated Frequency	$150.78Mhz$
Power supply	1.2V (pads and core)

TABLE I
F-TFR PROTOTYPE

We have just received the first prototype samples, which are currently under test.

[1] J. Ville, "Théorie et applications de la notion de signal analytique," *Cables et Transmission*, vol. 2A, pp. 61–74, 1948.

[2] L. De Vito, S. Rapuano, and G. Truglia, "A 3-D baseband signal analyzer prototype for 3G mobile telecommunication systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 4, pp. 1444–1451, Aug. 2005.

[3] M. Saab, "Interpretation de la PFT comme distribution de Wigner. application à la détérmination de la fréquence et de la phase instantanée d'un signal HF modulé," M.S. thesis, Supelec, 2004.

[4] RF Engines Ltd, "Pipelined frequency transform," `http://www.rfel.com/products/Products_pft.asp`.

[5] I. Koren, *Computer Arithmetic algorithms*, pp. 146,147, AK Peters, 2002.

[6] L. Noury, H. Mehrez, F. Durbin, and A Tissot, "Use of multiple numeration systems for architecture and design of a high performance FIR filter netlist generator," in *The 16th International Conference on Microelectronics*, 6-8 Dec. 2004, pp. 547–550.

[7] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 1, pp. 349–356, march 1965.

[8] J. Fadavi-Ardekani, "Booth encoded multiplier generator using optimized Wallace trees," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 1, no. 2, pp. 120–125, June 1993.

[9] A. Greiner and F. Pecheux, "Alliance: A complete set of CAD tools for teaching VLSI design," in *Third EuroChip Workshop*, 1992.

[10] S. Belloeil, J.-P. Chaput, R. Chotin-Avot, C. Masson, and H. Mehrez, "Stratus : un environnement de développement de générateurs d'IP numériques," in *9èmes Journées Pédagogiques du CNFM*, Rennes, France, 2006.

[11] C. Alexandre, H. Clement, J.-P. Chaput, M. Sroka, C. Masson, and R. Escassut, "Tsunami: an integrated timing-driven place and route research platform," in *Design Automation and Test in Europe 2005*, 2005, vol. 2, pp. 920–921.