International Journal of Electronics Signals and Systems

Volume 2 | Issue 3 Article 5

January 2013

DESIGN AND IMPLEMENTATION OF ASYNCHRONOUS FIR FILTER

N. HIMABINDU
Dept. of ECE, Reva ITM, Bangalore, India- 560064, raju.himabindu@gmail.com

ROHINI DESHPANDE
Dept. of ECE, Reva ITM, Bangalore, India- 560064, rohini_deshpande@revainstitution.org

DESIGN AND IMPLEMENTATION OF ASYNCHRONOUS FIR FILTER

[1]**HIMABINDU N & [2]ROHINI DESHPANDE**

[1&2]Dept. of ECE, Reva ITM, Bangalore, India- 560064
E-mail : raju.himabindu@gmail.com, rohini_deshpande@revainstitution.org

Abstract - This paper presents the architecture of a micropipeline asynchronous digital signal processing chain coupled to non-uniformly sampled data in time. Non-uniform sampling has been proven to be a better scheme than the uniform sampling to sample low activity signals. With such signals, it generates fewer samples, which means less data to process and lower power consumption. In addition, it is well-known that asynchronous logic is a low power technology. We focus on a Finite Impulse Response filter (FIR) applied to this non-uniform sampled signal obtained from an asynchronous analog to digital converter (A-ADC). The FIR filter blocks are implemented using verilog code.

Keywords *- Asynchronous logic, non-uniform sampling, Level-crossing sampling, FIR filter.*

I. INTRODUCTION

With the enormous growth in multimedia applications, the need for low power and high-performance digital signal processing (DSP) devices is higher than ever. One of the most widely used operations performed in DSP is *finite impulse response* (FIR) filtering. FIR filter performs the weighted summations of input sequences, which are frequently used to implement highpass, lowpass, and many other types of filters.

With the increasing system on chip complexity, several problems become more and more critical, and affect severely the performance of the system. These issues can have different form such as: power consumption, clock distribution, electromagnetic emission, etc. Synchronous logic seems to reach its technological limits when dealing with these problems. However, asynchronous logic has proven that it could be a better alternative in many cases. It is well known that it has many interesting properties such as immunity to metastable states [2], low electromagnetic noise emission, low power consumption, high operating speed or Robustness towards variations in supply voltage, temperature, and fabrication process parameters.

Moreover, non-uniform sampling and especially the level crossing sampling become more interesting and beneficial when they deal with specific signals like temperature, pressure, electro-cardiograms or speech which evolve smoothly or sporadically. Indeed, these signals are able to remain constant on a long period and to vary significantly during a short period of time.

Therefore using the Shannon theory for sampling such signals leads to unwanted samples which increases artificially the computational load. Classical and uniform sampling takes samples even if no change occurs in the input signal.

A new class of ADCs, called asynchronous ADCs (AADCs) has been developed by the TIMA Laboratory [7]. This A-ADC is based on the combination of a level-crossing sampling scheme and a dedicated asynchronous logic [5]. The asynchronous logic only samples digital signals when an event occurs, i.e. a sample is produced by the A-ADC which delivers non-uniform data in time. This event-driven architecture combined with the level-crossing sampling scheme is able to significantly reduce the dynamic activity of the signal processing chain.

This paper presents an asynchronous FIR filter architecture, based on a micro-pipeline asynchronous design style. Section II describes the asynchronous logic and more precisely about micro-pipeline asynchronous circuit. Section III describes the A-ADC architecture and also the non-uniform sampling scheme. Section IV presents the asynchronous FIR Filter architecture. Finally conclusions are drawn in section V.

II. ASYNCHRONOUS LOGIC

Asynchronous logic is well known for its interesting properties that synchronous logic doesn’t have: such as low Electromagnetic emission, low power consumption, robustness, etc. [1]. It has been proved that this logic improves the Nyquist ADCs performances in terms of immunity to metastable states [2], low electromagnetic emission [3] or low power consumption [4]. This section briefly presents the main asynchronous logic principles. It also describes how asynchronous micro-pipelined circuits are build, from two distinct parts - the data path, and the asynchronous control path.

Unlike synchronous logic, where the synchronization is based on a global clock signal, asynchronous logic doesn't need a clock to maintain the synchronization between its subblocks. It is considered as a data driven logic where computation occurred only when new data arrived. Each part of an asynchronous circuit, establishes a communication protocol, with its neighbors in order to exchange data with them. This kind of communication protocol is known as "hand shake" protocol. It is a bidirectional protocol, between two blocks called Sender and Receiver as showed in Figure 1

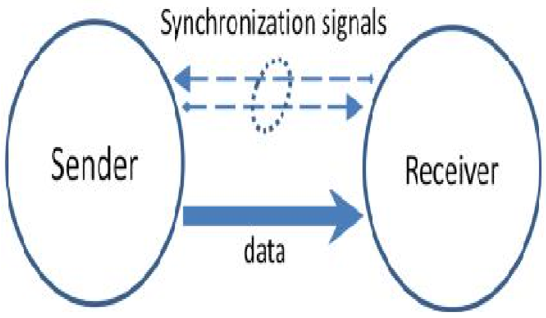

Fig. 1: Handshake protocol is established between two subblocks of an asynchronous circuit that need to exchange data between each other

The sender starts the communication cycle by sending a request signal "req" to the receiver. This signal means that data are ready to be sent. The receiver starts the new computation after the detection of the "req" signal, and send back an acknowledge signal "ack" to the sender marking the end of the communication cycle, so a new one could start. The main gate used in this kind of protocol, is the "Muller" gate or also known as C-Element. It helps to detect a rendezvous between different signals. The C-element is a state-holding gate.

A. *Micropipeline Circuit*

One of the most known asynchronous logic styles is the micropipeline style. The choice of the asynchronous style affects the circuit implementation. Among all the asynchronous circuit styles the micro-pipeline has the most closely resemblance with the design of synchronous circuits due to the extensive use of timing assumptions [5]. Same as a synchronous pipeline circuit, the storage elements is controlled by control signals. Nevertheless, there is no global clock. These signals are generated by the Muller gates in the pipeline controlling the storage elements.

A simple asynchronous micro-pipeline circuit, could be built by using transparent latches as storage elements as shown in Figure 2.

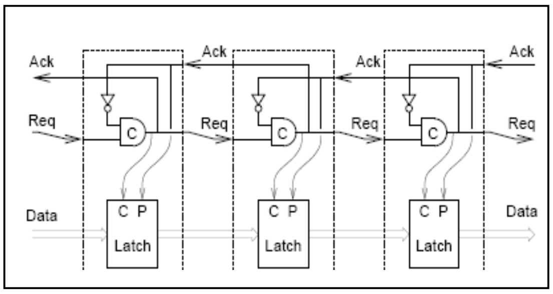

Fig. 2: Micropipeline asynchronous circuit

III. ASYNCHRONOUS ANALOG TO DIGITAL CONVERTER – 'AADC'

The asynchronous analog-to-digital converter (A-ADC) presented in [7] is based on an irregular sampling scheme called level-crossing sampling [6] and an asynchronous implementation. The system is only driven by the information present in the input signal. It only reacts to the analog input signal variations.

A. *Irregular Sampling*

For irregular sampling and a M-bit resolution, 2M-1 quantization levels are regularly disposed along the amplitude range of the signal. A sample is taken only when the analog input signal i(t) crosses one of them. Contrary to classical Nyquist sampling, the samples are not regularly spaced out in time, because they depend on the signal variations. Thus, together with the value of the sample i_n, the time dti_n elapsed since the previous sample i_{n-1} must be recorded. A local timer of period T_C is dedicated to this task. The amplitude of the sample is precise, but the time elapsed since the previous sample is quantized according to the precision T_C of the timer. The SNR depends on the timer period T_C, and not on the number of quantization levels.

Thus, for a given implementation of the irregular sampling A/D converter, the SNR can be externally tuned by changing the period T_C of the timer. In theory, for irregular sampling, the SNR can be improved as far as it is needed, by reducing T_C. The limit is the accuracy of the analog blocks: they determine the precision of the quantization levels position in Figure 3.

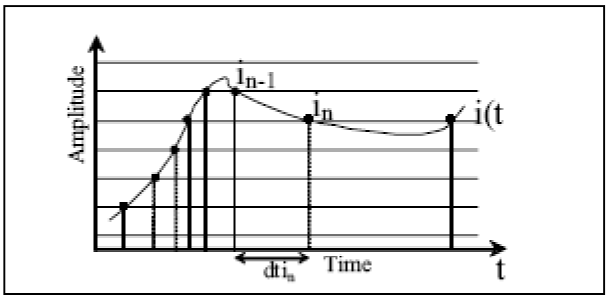

Fig. 3: Level crossing Sampling

B. Asynchronous ADC

The block diagram of the architecture performing this sampling scheme is shown in Figure 4. The converter resolution M and its dynamic range Δ set the quantification step:

$$q = \frac{\Delta}{2^M - 1}$$

The output digital value V_{num} is converted to V_{ref} by the DAC, and compared to the input signal i(t).

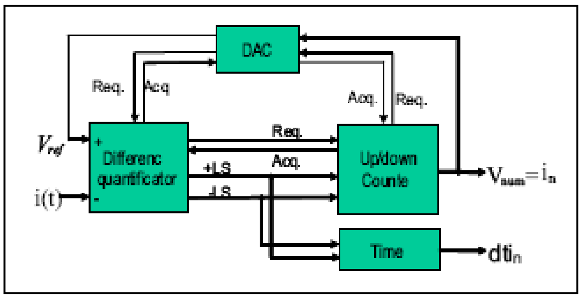

Fig. 4: Block diagram of the A-ADC

If the difference is greater than ½q, the counter is incremented, if it is lower than -½q, it is decremented. In all other cases, nothing is done. The converter output signal V_{num} remains constant, there is no activity. The output signal is composed of couples (i_n, dti_n) where i_n is the digital value of the sample, and dtin the time elapsed since the previous converted sample in-1, given by the timer. Information transfer is locally managed with a bi-directional control signaling. Each “data” signal is associated with two “control” signals: a request and an acknowledgement [1]. Let δ be the total delay of the loop, the slope of i(t) must satisfy the “tracking condition”:

$$\left|\frac{di(t)}{dt}\right| \leq \frac{q}{\delta}.$$

In digital signal filtering, an input signal i(t) represented by its samples (i_n, dti_n) is processed to obtain an output signal o(t) represented by its samples (o_n, dto_n).

IV. ASYNCHRONOUS FINITE IMPULSE RESPONSE FILTERING

This section formally defines the FIR filtering computation when processing irregularly sampled digital signals.

A. Principle

In a regular scheme, an Nth order FIR filter computes a digital convolution product, where T_{sample} is the sampling period, as described in Eq. below.

$$o(t_n) = T_{sample} \sum^{N} h(t_k) i(t_{n-k})$$

Contrary to the regular scheme, the sampling time of the kth sample of the impulse response h does not necessarily correspond to the sampling time of the (n-k) th sample of the input signal i.

$$th_k \neq ti_{n-k}\,.$$

The product of the two samples is thus meaningless. The difficulty is bypassed by processing the convolution product between the resampled interpolated impulse response hˆ and the resampled interpolated input iˆ. Indeed, the output of a filtering process is defined as the numerical result of an analog convolution product of two analog signals obtained from two irregularly sampled signals by an interpolation. Actually, the convolution product is an area computation. It is well known that the easier way to compute this area is the rectangle method i.e. a zero-order method. Moreover in the classical digital computation, the output is also an area computation using the rectangle method, but all the intervals are constant, which it is not the case in irregular sampling.

This means that it is necessary:

- to interpolate both input and impulse response signals at order 0,
- to resample the impulse response signal at the input sampling times,
- to resample the input signal at the impulse response sampling times.

At each new input sampling step, two new sampled signals hˆ and iˆ are calculated. Let n hˆ and n iˆ be the new irregular series of samples used to process the nth output sample. The time of this sample corresponds to the delay introduced in the computation. The output is computed at the times of the input samples, which could be at any time. Then, the output signal is given by the following eq.

$$\begin{cases} o_n = o(ti_n) = \int \hat{h}_n(\tau)\hat{i}_n(ti_n - \tau)d\tau, \\ dto_n = dti_n. \end{cases}$$

Figure 5 shows the general principle of resampling for the input signal and the impulse response of the filter. The continuous lines represent the original samples whereas the dashed lines correspond to the new calculated samples. After this step, each sample of $\hat{i}_n$ corresponds to a sample of $\hat{h}_n$. It is now possible to process the convolution product.

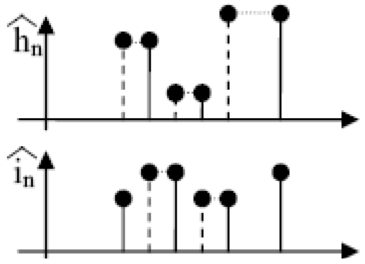

Fig. 5: Principle of the resampling scheme used in the irregular FIR computation.

B. *Micropipeline asynchronous FIR filter implementation*

The structure as in figure 5 describes the architecture of the FIR filter

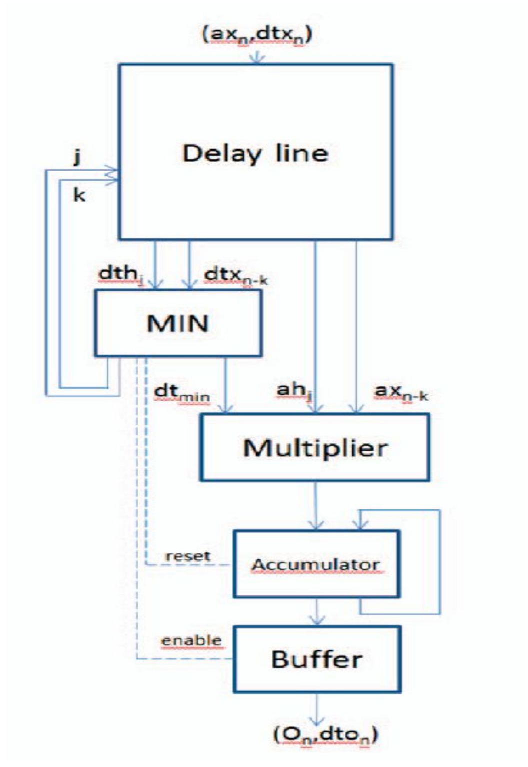

Fig. 6: Iterative structure for the Asynchronous FIR Filter samples.

The DELAY LINE block is the memory of the FIR filter. It is a shift register that stores the input samples. The “Delay Line” gets the sampled signal data (ax_n, dtx_n), from the A-ADC. The communication between these two blocks is based on the handshake protocol.

The MIN block is in charge of computing indices k and j as well as the current minimum time interval (dt_{min}). To this aim, the MIN block handles two local variables for k and j, a local variable for the current time interval (dt_{n-k} or dth_j) and the T_{sample} constant value.

MIN generates at the end of each convolution product cycle, a “reset signal and an “enable signal to reset the output of the Accumulator and enable the output of the Buffer.

Then the “Multiplier” computes all sub-areas value (dt_{min} $*ax_{n-k}*ah_j$) that are accumulated in the “Accumulator” in order to compute the convolution product.

V. CONCLUSION

An asynchronous FIR Filter architecture is presented in this paper, along with an asynchronous Analog to digital converter (A-ADC). The proposed FIR Filter architecture is designed using the micro-pipeline asynchronous style. The filter blocks are implemented using verilog code.

REFERENCES

[1] M. Renaudin, “Asynchronous Circuits and Systems: a Promising Design Alternative”, Journal of Microelectronic Engineering, Vol. 54, pp. 133-149, 2000.

[2] D. Kinniment et al., “Synchronous and Asynchronous A-D Conversion”, IEEE Trans. on VLSI Syst., Vol. 8, n° 2, pp. 217-220, April 2000.

[3] D.J. Kinniment et al., “Low Power, Low Noise Micropipelined Flash A-D Converter”, IEE Proc. On Circ. Dev. Syst., Vol. 146, n° 5, pp. 263-267, Oct. 1999.

[4] L. Alacoque et al., “An Irregular Sampling and Local Quantification Scheme A-D Converter”, IEE Electronics Letters, Vol. 39, n° 3, pp. 263-264, Feb. 2003.

[5] “Principles of Asynchronous Circuit Design – A Systems Perspective”, Edited by JENS SPARSØ Technical University of Denmark & STEVE FURBER The University of Manchester UK.

[6] J.W. Mark et al., “A Nonuniform Sampling Approach to Data Compression”, IEEE Trans. on Communication. Vol. COM-29, n° 4, pp. 24-32, Jan. 1981. W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[7] F. Aeschlimann, E. Allier, L. Fesquet, M. Renaudin, "Asynchronous FIR Filters: Towards a New Digital Processing Chain," Asynchronous Circuits and Systems, International Symposium on, pp. 198-206, 10th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'04), 2004.

◈◈◈