2006

Implementation of a quasi-digital ADC on PLD

Fu-yuan Wang
Zhengzhou University

Yong-liang Li
Zhengzhou University

Jiangto Xi
Faculty of Informatics, University of Wollongong, jiangtao@uow.edu.au

Joe F. Chicharo
University of Wollongong, chicharo@uow.edu.au

Publication Details

Wang, F., Li, Y., Xi, J. & Chicharo, J. F. (2006). Implementation of a quasi-digital ADC on PLD. Proceedings of 2006 8th International Conference on Solid State and Integrated Circuit Technology (pp. 1791-1793). Piscataway, USA: IEEE.

Implementation of a quasi-digital ADC on PLD

Abstract

This paper presents a new way to implement stochastic logic-based analog-to-digit converters (ADCs) on a programmable logic device (PLD) chip. The proposed implementation is almost all digitalized so that the design can be done by using hardware description language and the results can be easily downloaded into a PLD chip. Both simulation and hardware test results are given.

Disciplines

Physical Sciences and Mathematics

Publication Details

Wang, F., Li, Y., Xi, J. & Chicharo, J. F. (2006). Implementation of a quasi-digital ADC on PLD. Proceedings of 2006 8th International Conference on Solid State and Integrated Circuit Technology (pp. 1791-1793). Piscataway, USA: IEEE.

Implementation of a Quasi-digital ADC on PLD

Fu-yuan Wang[1], Yong-liang Li[1], Jiang-tao Xi[2], Joe F. Chicharo[2]
1. College of Information Engineering, Zhengzhou University, Henan, 450002, China
2. The University of Wollongong, NSW, Australia
Email: Wangfy@zzu.edu.cn , Yongliangli313@163.com, Jiangtao@uow.edu.au

Abstract

This paper presents a new way to implement stochastic logic-based Analog-to-digit converters (ADCs) on a Programmable Logic Device (PLD) chip. The proposed implementation is almost all digitalized so that the design can be done by using hardware description language and the results can be easily downloaded into a PLD chip. Both simulation and hardware test results are given.

1. Introduction

Digit-to-Analog Converter (DAC) and Analog-to-Digit Converter (ADC) are commonly used components in electronic systems. As they contain both analog and digital circuits, design, implementation and fabrication of them are usually more costly and difficult. In systems involving data acquisition, ADCs are usually at the front part and the remaining part is usually all digital. In this situation digital implementation of ADC will make it possible to integrate the whole system into a single PLD chip or digital Application Specific Integrated Circuits (ASIC). For this reason extensive efforts have been made on developing digitalized DAC and ADC[1][2][3]. However, there is not such product in the market so far.

In 1995 Ortega [1] proposed a method to implement 'quasi-digital' ADC and DAC using stochastic logic. The proposed approach was realized as ASIC chips which are almost all digital although it contains a couple of analog components. Due to the advantages of PLD in contrast to ASICs, such as short design period, enhanced flexibilities and reduced risk of fabrication, we have implemented the stochastic logic-based ADCs on PLD chips and this paper presents detailed of the work.

This paper is organized as follows. Section 2 gives the principle of stochastic ADC proposed by Ortega [1]. In Section 3 we present the details of the design and implementation of the stochastic ADC using PLDs. The performance of the proposed ADC was tested using different implementations, and the results are shown in Section 4. Finally Section 5 concludes the paper.

2. The principle of stochastic ADC

The stochastic ADC is based on the stochastic logic theory proposed by Gaines in 1969[4]. The stochastic logic was proposed for DAC as follows. A binary number is converted to a sequence of constant level pulses in such a way that the number of high level pulses is proportional to the value of the binary number. As the positions of high level pulses are stochastic, the resulting pulses sequences are called stochastic logic pulse sequence. Low-pass filtering of the stochastic sequence will yield the analog signal level of the results. An approach for converting digital numbers to such pulse sequence was proposed in [5], based on which Ortega [1] constructed DAC and ADC and implemented them as ASIC chips, which is shown in Figure 1. The left side of the doted line is the analog part including an RC low pass filter and an analog comparator while the right side is digital part including a successive approximation register (SAR) and a circuit which transfers digital number to stochastic pulse sequence.

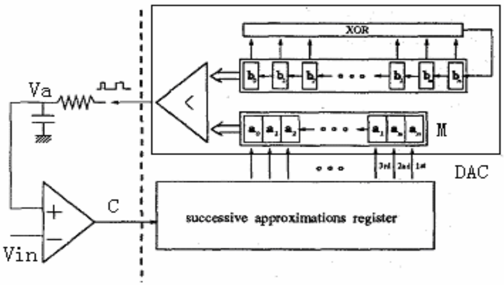

Figure 1. ADC based on stochastic logic

The working principle of Figure 1 is described as follows. V_{in} is the analog voltage to be converted which is connected to the inverting input of the comparator. The other input of the comparator V_a is a voltage rebuilt from the stochastic logic sequence after it passes through a low pass filter. While the system is initialized, the Most Significant Bit (MSB) of SAR must be set to '1', and be read to digit register M simultaneously. The number in M is transferred into a sequence of stochastic logic pulse. When the pulses pass through the low pass RC filter, V_a is obtained.

After V_a is obtained, it is then compared with V_{in}. The

1-4244-0161-5/06/$20.00 ©2006 IEEE

output of the comparator is signal C that controls the output of SAR. When $V_a \geq V_{in}$, it means the number in M is too big, the analog comparator outputs '1', namely C equals to 1. Then C acts as a control signal which clears the MSB bit of SAR and sets the next MSB bit. Otherwise, the analog comparator outputs '0'. In this case, SAR keeps its MSB unchanged and sets the next MSB. The comparison will continue until the Least Significant Bit (LSB) of SAR is set and the final numbers in SAR are the digit numbers corresponding to the analog voltage V_{in}.

3. PLD Realizations

In order to implement a stochastic ADC using PLD, the following problems must be solved. (1)Timing must be synchronized between analog comparator and SAR; (2) V_a must keep stable when comparison is made, which requires that low pass filter and the digital part are synchronized in terms of timing; (3)Implementation of circuit for digital to stochastic pulse conversion; (4) Implementation of SAR. We will provide the solutions for those issues.

Similar to that in Figure 1, the ADC proposed contains only one analog comparator LM393, one resistor and one capacitor. The other parts of the ADC are all digital circuits that can be wholly downloaded into PLD chip. Figure 2 gives the structure of an 8-bit ADC that contains three modules SAR_8, SAR_CLK and Comp_1 based on stochastic logic.

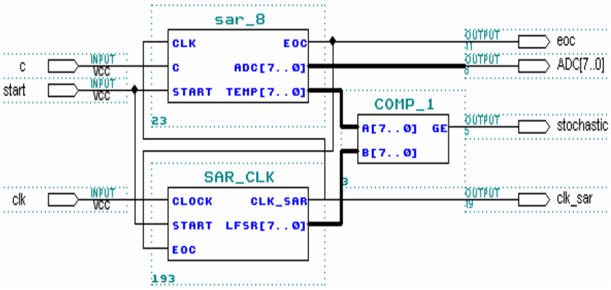

Figure 2. The module structure of ADC

SAR_8 is to realize the SAR and latch the final digital number. After the final number is obtained, it gives a signal EOC that indicates the completion of conversion. The input clock signal of this module is provided by SAR_CLK. C is the output of analog comparator, START is the starting signal of conversion, EOC is a signal for end of conversion and ADC[7..0] is the final digital output after conversion while Temp[7..0] is temporally memorized digital number.

SAR_CLK is used as a generator of stochastic numbers producing stochastic numbers according to [5] and signal CLK_SAR acting as a clock input for SAR_8. In this module LFSR[7..0] is the stochastic numbers produced. For an 8-bit ADC the period of LFSR[7..0] is 255[5].

In order to achieve more precise timing matching between SAR_8 and CLK_SAR, a counter is employed to compensate for the RC delay, ensuring that stochastic pulse sequences are long enough passing the low pass filter so that V_a is in steady condition. The counter increases 1 automatically after a period of LFSR[7..0]. By setting the maximum number of the counter, the long enough pulse sequences can pass RC filter and the steady V_a are ensured. Thus, we get correct variations of signal C, which have critical impact on SAR. In figure 2, there are also hand-shaking signals between module SAR_8 and SAR_CLK. One is signal EOC that is given after a conversion is finished and the other is CLK_SAR, which ensures the whole system to work coordinately after steady V_a is obtained.

Module Comp_1 realizes the function of digit comparison as shown in Figure 1 being part of digital to stochastic conversion circuit. The two inputs are Temp [7..0] from module SAR_8 and LFSR[7..0] from module SAR_CLK and the output is 'stochastic'. When Temp [7..0]⩾LFSR[7..0], it outputs high level 1, otherwise low level 0.

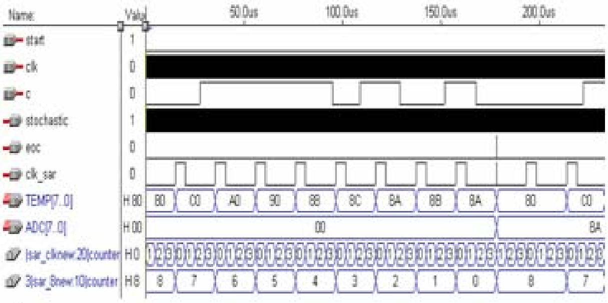

Figure 3. Timing simulation of Figure 2

Figure 3 is a simulation result of Figure 2. The timing could be seen from Figure3. When Start=1, the MSB of Temp[7..0] (here it is 80) shown in Figure 3 are set to 1, then the comparisons are made between LFSR[7..0] and Temp[7..0] on every rising edge of CLK and outputs signal 'stochastic'. For observation convenience, the maximum number of the counter in SAR_CLK is set to 3. For every zero appeared in the counter, CLK_SAR outputs 1.At the rising edge of CLK_SAR, Temp[7..0] is set again depending on C's condition (1 or 0). If C=1, for example, the MSB of Temp[7..0] is cleared and the second MSB bit is set to 1 at the second rising edge of CLK_SAR, thus the TEMP[7..0] changes its value from C0 to A0. When C=0, the MSB of Temp[7..0] keeps unchanged, the second MSB bit is set to 1. For example, at the fifth rising edge, Temp[7..0] changes its value from 88 to 8C. At the eighth rising edge, the LSB bit is fulfilled and at the ninth rising edge, the converted number is latched. At the same time, ADC[7..0] is output and EOC gives a high level lasting for a clock period. EOC is also the starting signal for next conversion to keep the conversions running continuously. As shown in Figure 3, when EOC=1, the MSB of Temp[7..0] is set to 1 again waiting for clock to start the next conversion.

4. Hardware Tests

We carried out experiment to test the performance. Two factors have to be considered while choosing R and C. Firstly, the bigger the product of RC, the longer the time needed for one conversion. Secondly, RC is the main part of low pass filter and we should make sure that the LSB of the digit number after conversion is correct. Formula (1)[1] is the selection rules of R and C in which $f_{\max}$ is the cutoff frequency, n the resolution and f_{clk} the clock frequency of SAR.

$$f_{\max} < 2^{-(2n-1)} f_{clk} \qquad (1)$$

The experiments have shown that the value by (1) is a conservative estimate. It usually gives bigger R and C so that makes longer conversion time. This paper gives a looser estimate by using simulation. The new way can reduce the conversion time while keeping the correction of conversion.

In our experiment the max analog voltage is 3.72V, and hence the correspondence LSB stands for analog voltage 14.50mV. In this case we should choose RC so that the fluctuation of V_a is less then 14.50mV. Simulation of Figure 4 using Protel99se has been done to observe the output of the filter shown in Figure 5. The Voltage source is 200 KHz rectangular pulse with 1:2 duty ratios. We can see from Figure 5 that, when RC is not properly selected, the output will exhibit big fluctuations.

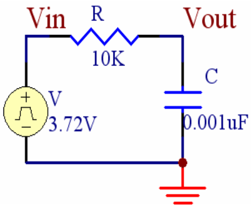

Figure 4. RC filter

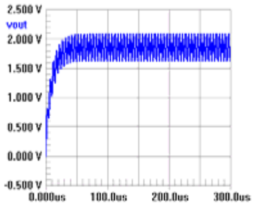

Figure 5.Big fluctuation output

The experiments are based on EPF10K10LC84-4 chip produced by Altera. The hardware simulation conditions are as follows: the analog comparator is LM393, clock frequency is 12 MHz, C＝0.001uF, the result tells that R can be as low as 20K while keeping the fluctuation less than 14.50mV. This result which is 0.616ms (sampling rate 1.62K/s) leads to less conversion time than (1).

In terms of practical implementation, the design should be optimized. It is expected that with EDA tool, the final design will occupy 63 logic cells (10% of the chip, the total number 576). The max working frequency of the system is 84.03MHz.

5. Conclusions

We have presented a PLD realization of quasi-digit ADC based on stochastic logic. It is almost all digital and only needs a couple of analog components. Experiments show that it is suitable for low sampling rate applications. Because the design is described by VHDL language, it is convenient to do modification and upgrading. Without changing the peripheral components, ADCs with different resolution can be realized by modifying the VHDL description.

References

[1] J.G. Ortega and C.L. Janer, “Analog to Digital and Digital to Analog Conversion Based on Stochastic Logic,” IEEE International Conference on Industrial Electronics, Control and Instrumentation, 1995, Vol.2, p.995-999.

[2] Takamoto and Mizuno, “An all-digital analog-to-digital converter with 12-uV/LSB using moving -average filtering,” IEEE Journal of Solid-State Circuits, Vol. 38, No.1 January, 2003.p.120-125.

[3] Toral,S.L and Quero, “Stochastic A/D sigma-delta converter on FPGA”,1999 IEEE 42nd Midwest Symposium on Circuits and Systems, Aug 08-Aug 11 1999, Las Cruces, NM, USA, Midwest Symposium on Circuits and Systems,Vol.1,1999, p.35-38.

[4] B.R.Gaine. “Stochastic Computing System” Advances In Information Systems Science 2.pp37-172, 1969.

[5] C.L.Taner and J.M.Quero, “Fully Parallel Summation in a New Stochastic Neural Network Architecture”, IEEE Int. Conf. on Neural Net, p.1498-1503, San Francisco, 1993.