Minsk, Belarus, October 2-4, 2001

FPGA BASED REAL-TIME IMAGE PROCESSING PLATFORM

M. Pieper, A. Kummert

Communication Theory, Department of Electrical and Information Engineering, University of Wuppertal, Rainer-Gruenter-Strasse 21, 42119 Wuppertal, Germany, phone. +49-202-4391961, email: mpieper@uni-wuppertal.de

ABSTRACT

In this paper, a FPGA based Real-Time Image Processing Platform is presented. In other words, the main processing of the digital input data is done in an FPGA. Signals up to a frequency of 40 MHz can be processed. The sampling rate can be adapted to the particular signal. One of the main advantages of the board is its ability to restart the FPGA without the help of an external computer.

1. INTRODUCTION

Over the past years programmable chips like field programmable gate arrays (FPGA), complex programmable logic devices (CPLD) and application specific integrated circuits (ASICs) became more and more important for digital image processing. One of the main benefits when using these devices is the ability of perfectly parallel signal processing. The board is capable of processing analog signals up to a maximum frequency of 40 MHz. The board can be separated into four different parts, namely clock-management, in-/ output sections, memory, and digital part.

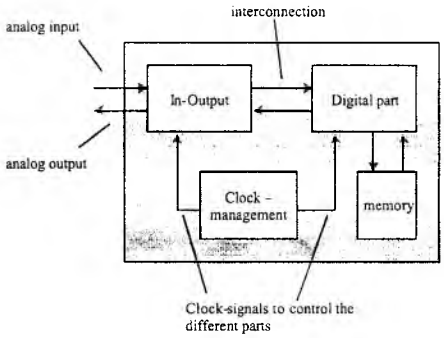

Fig. 1 Block diagram of the raw structure of the FPGA based Image Processing-Board

Figure 1 shows a block diagram of the four parts. The FPGA concept admits highly variable signal processing schemes. For example, if a modified hardware implementation or an update of an already given algorithm is required, this situation can be handled without any problems. One of the basic features of any FPGA solution is the fact that implemented data is lost when the chip is detached form the voltage supply. For applications that do not require a common update of the implemented logic, the board provides the possibility to restart the FPGA by means of a program, loaded from a FlashROM. To realise this functionality, three different modes to restart the board are provided. The most common solution is based on reloading the configuration data into the FPGA by means of a personal computer via a programming interface. The second option is to additionally write in parallel the data into an SRAM. After terminating the configuration of the FPGA, the data of the SRAM is copied into a FlashROM. It is very important to store the data in this way, because ROMs cannot be programmed as fast as FPGAs. The third possibility to start up the FPGA without a computer, relies on the use of the memorized data from the FlashROM.

2. IN- AND OUTPUT SECTION

The in- and output sections of the board are responsible for a correct recording and dispensing of the incoming and processed signals. Main task of the input section is to digitalize the analog input signals. The latter task is realized by means of an analog-to-digital converter (ADC) [3]. The sampling frequency of the ADC can likewise be selected between 5 MHz and 80 MHz. A pre-processing of the analog input data is useful to

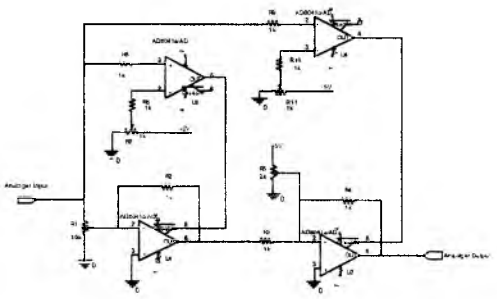

Fig. 2 pre-amplification section

cover the whole input range of the ADC. Fig. 2 shows the circuit of the pre-amplification.

Op-amps U1 and U2 amplify the analog input signal. Both are used as inverting amplifiers. U1 amplifies the input signal to cover the whole input range of the ADC. U2 is needed to invert the amplified signal. The other op-amps (U3, U4) are operated as comparators to compare the input signal to predefined thresholds. Their output is connected to the disable-inputs of the op-amps U1 and U2 which are responsible for the preconditioning of the input signal. If the input signal crosses one of these thresholds the amplification of the op-amps is reduced for the moment of the transgression.

The following figure shows a comparison between results from a PSpice-simulation (Fig.3a) and a real measurement (Fig.3b) recorded by means of a four-channel-oscilloscope which demonstrates the protection of the ADC. It should

Fig. 3a Simulation of the protection circuit from Fig.2

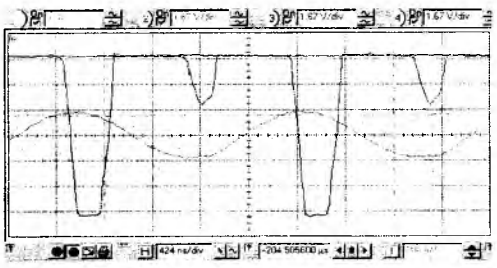

Fig. 3b Measurement of the protection circuit from Fig.2

be noted, that - in contrast to the simulation - the used oscilloscope can only illustrate the input-, the two output- and one of the two threshold-signals. The signals processed by the FPGA can optionally be used in a digital form for further processing or digitally stored or be converted back into an analog signal. The analog output signal is provided by an analog to digital converter (DAC).

3. CLOCK-MANAGEMENT

The clock-management is used to control all digital devices of the board. The main clock signal is generated by a 80 MHz quartz oscillator. By means of a CPLD, the sampling rate of the ADC can optionally be chosen arbitrarily in the range of 5 MHz up to 80 MHz. For an accurate sampling of the analog input signal, it is useful to adapt the sampling rate to different input signal classes. For example, signals in the frequency range up to 2 MHz do not need to be sampled with a sampling rate of 80 MHz. For these kinds of signals sampling rates of 4 MHz or 5 MHz are completely sufficient. Another benefit of the clock-management can be seen in the fact, that ADC and FPGA can be operated at different frequencies. The clock rate of the FPGA can likewise be selected in the same range as for the ADC. Obviously, this is only meaningful if the sampling rate of the FPGA is higher than or equal to the sampling rate of the ADC. This feature allows to process a digitalized input signal with a clock-rate higher than the sampling frequency. This behaviour is advantageous if complex asynchronous algorithms have to be implemented. The different clock-rates for the ADC and the FPGA are generated by means of a counter. Each rising edge of the clock signal is detected and the counter is incremented. If the counter reaches a defined value it will be set back to zero to prevent a buffer overrun. Using the counter's values, the different clock signals can be generated, as illustrated by Fig. 4a and Fig. 4b.

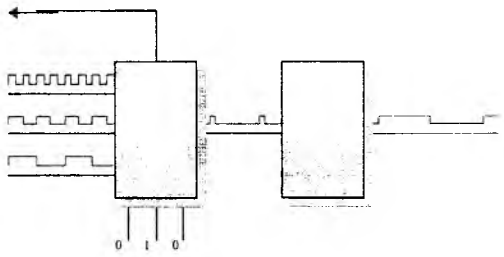

Fig. 4a Generation of the sampling rate of the ADC

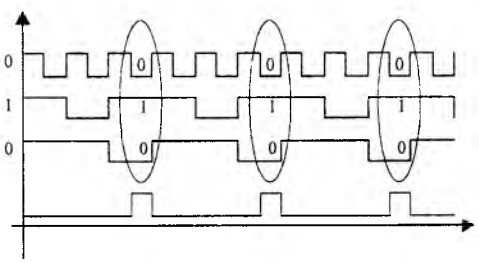

Fig. 4b Conformity of the reference signal and the generated clock signals

The 80 MHz clock is reduced to a 40 MHz clock and used to drive the CPLD, because the implanted logic of the CPLD can be driven by a maximal frequency of up to 45 MHz. The 40 MHz clock is divided two times, generating three new clocks with different frequencies (20 MHz and 10 MHz). These signals and the 40 MHz clock are compared to a 3-bit reference signal. If the values of the three clocks match the reference signal, an impulse is generated. The latter is used to restart the counter. The sampling clock of the ADC or the clock of the FPGA is generated by converting this impulse signal into a clock-signal

4. DIGITAL SECTION

As described above, the main task of the board is to process digital signals. To enable this, the application specific signal processing algorithm has to be implemented in the FPGA. This can be done

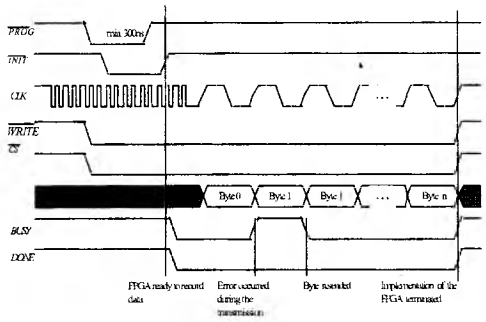

Fig. 5 Configuration clocking sequence

by using one of the three described modes. To operate an FPGA, eight different signals have to be generated or controlled. Fig. 5 shows a wave diagram of the initialization phase of an FPGA [4][5][6].

After power on, the configuration memory of the FPGA is automatically cleared. The $\overline{INIT}$ pin transitions to High when the clearing of the configuration memory is complete. A logical Low on the $\overline{PROG}$ pin resets the configuration memory and holds the FPGA in the clear configuration memory state. A minimum Low pulse of 300 ns is needed to clear the whole configuration memory. If the $\overline{INIT}$ pin switches to a logical High, the initialization of the FPGA is terminated and the chip is prepared to be implemented. The inputs $\overline{WRITE}$ and $\overline{CS}$ have to be set to a logical Low. A logical Low on the $\overline{DONE}$ pin indicates, that the configuration of the FPGA is not yet finished. The $\overline{BUSY}$ pin indicates errors during the transmission of the configuration bytes. If some error occurs the $\overline{BUSY}$ pin generates a logical High until the wrong byte is sent again and recorded correctly. After configuration, the $\overline{DONE}$ pin switches to logical High.

The most common solution is based on reloading the configuration data into the FPGA by means of a computer. To achieve this, the signals $\overline{WRITE}$, $\overline{CS}$ and $\overline{PROG}$ are generated by the computer. Hence, it is only important to ensure that no device e.g. the CPLD sends any signals to the FPGA.

To write the configuration data in parallel to a SRAM during the implementation of the FPGA in the second mode, the control signals for the SRAM [2] have to be generated by the CPLD. The end of the configuration of the FPGA is indicated with a logical High of the $\overline{DONE}$ pin. To make sure that the data can be copied to the FlashROM [1], the FPGA must be hold in the configuration status. This could be done be pulling the $\overline{DONE}$ pin to a logical Low. Afterwards it is ensured that the FPGA cannot use the SRAM to save data, and the configuration data of the SRAM can be written to the FlashROM. To enable this, the CPLD has to generate signals for the SRAM to read data and the FlashROM has to be instructed to memorize the data. The configuration data has to be saved in the SRAM first, because the FlashROM cannot be programmed as fast as the FPGA.

The last mode permits to restart the FPGA from the FlashROM, so that no external host computer is needed. To restart the FPGA with the memorized data, the CPLD has to generate all control signals for the FlashROM to send the data to it. To configurate the FPGA with the data of the FlashROM, the CPLD has also to generate the signals for the FPGA ($\overline{WRITE}$, $\overline{CS}$ and $\overline{PROG}$) and to supervise the signals $\overline{BUSY}$ and $\overline{DONE}$ in order to respond to potential errors or to detect the end of the configuration.

These three possibilities to start or restart the board are controlled by the CPLD. The program to control this procedure is written in the hardware description language VHDL. The three modes can easily be chosen by means of jumpers. After configuration of the FPGA, the SRAM can be used

to memorize calculation data of the FPGA. In this case the FPGA has to generate control signals for the SRAM.

5. CONCLUSION

In this paper, a stand-alone digital real-time image processing-board based on an FPGA has been presented. The board features the possibility to process analog signals in the frequency range of 2 MHz up to 40 MHz. The input signals can be sampled with different sampling rates to prevent redundant values. The synchronization of the ADC and the FPGA is no problem, because the ADC sends output data on the falling edge of the sampling clock. The clock of the ADC is used to trigger the FPGA and to receive data on the rising edge to secure that only stable values are processed by the FPGA.

The configuration data of the FPGA can be implemented in the common way by means of a computer. Furthermore, the board can be initialised without the need of external host computer access. This feature makes it best suited for evaluation of new signal processing algorithms.

REFERENCES

[1]. Atmel Corporation „4-Megabit (512 x 8) 3-volt Only 256-Byte Sector Flash Memory AT29LV040A" Rev: 0334D-10/98

[2]. Cypress Semiconductor „CY7C1049V33 512 x 8 Static RAM" Corporation June 2, 1999

[3]. Texas Instruments „TLV5580 8-BIT, 80 MSPS LOW-POWER A/D CONVERTER" SLAS205 – December 1998 – Revised January 1999

[4]. Xilinx „Configuring Virtex FPGAs from Parallel EPROMs with a CPLD" XAPP 137 March 1, 1999 (Version 1.0)

[5]. Xilinx „Virtex™ 2.5 V Field Programmable Gate Array Final Product Specification" DS003 (v.2.2) May 23, 2000

[6]. Xilinx „The Programmable Logic Data Book 1999" Xilinx Product Specification, 1999