

# High speed video recording system on a chip for detonation jet engine testing

Alexander N. Samsonov\*, and Khristina V. Samoilova

<sup>1</sup>Lavrentiev Institute of Hydrodynamics SB RAS, 630090 Novosibirsk, Russia

**Abstract.** This article describes system on a chip development for high speed video recording purposes. Current research was started due to difficulties in selection of FPGAs and CPUs which include wide bandwidth, high speed and high number of multipliers for real time signal analysis implementation. Current trend of high density silicon device integration will result soon in a hybrid sensor-controller-memory circuit packed in a single chip. This research was the first step in a series of experiments in manufacturing of hybrid devices. The current task is high level syntheses of high speed logic and CPU core in an FPGA. The work resulted in FPGA-based prototype implementation and examination.

## 1 Introduction

High speed video recording device bases on a sensor equipped with a large number of parallel data buses, each of which calls "channel". Data is transmitted over up to 16 channels in parallel clocked by the same clock wire. The channel width is equal to ADC output width which is usually 10 or 12 bits. However, common data width largely exceed 160 bits and it does not allow to use general purpose CPUs.

The first prototype was constructed using FPGA, because it is suitable for debugging and verifying of architecture and firmware. During architecture development the number of FPGA in device was increased to four. Each of FPGA conducts up to four 12 bit data channels and it controls external DDR3-memory. Incoming data is transmitted through internal FIFOs to realize cross domain clock synchronizations. There are a lot of clock domains because data width varies in different interfaces. For example memory width is 24-bit, external Gigabit interface width is 8-bit and input data width is up to 72 bit for single FPGA.

## 2 Arithmetic

High-speed video recording process is a memory intensive routine, which demands real time data analysis. Also sensor calibration procedure is performed during data analysis. Calibration coefficients are stored in internal memory. Normalized and linearized data is fed through outline recognition module and image recognition is performed.

---

\* Corresponding author: [info@ck-llc.com](mailto:info@ck-llc.com)

Implemented core includes following arithmetic and logical functions tested in FPGA and applied for image recognition [1]:

- averaging;
- differentiating;
- segmentation;
- hysteresis based on previous frame and threshold level;
- pixel data confidence.

Some of these algorithms demand too many multipliers. Some of them are memory intensive. This property was important during FPGA family selection. So multipliers in Xilinx Virtex 5 was organized in columns (in sequence of) 64 pcs, but delay between different columns if you connect it serially was too big [2]. Altera FPGA does not have this problem: all off multipliers in Stratix V can be connected serially. But, nevertheless, number of multipliers depends on FPGA family and price. ASIC implementation will help create appropriate sequence of multipliers connected in according to minimal delays.

### **3 FPGA architecture**

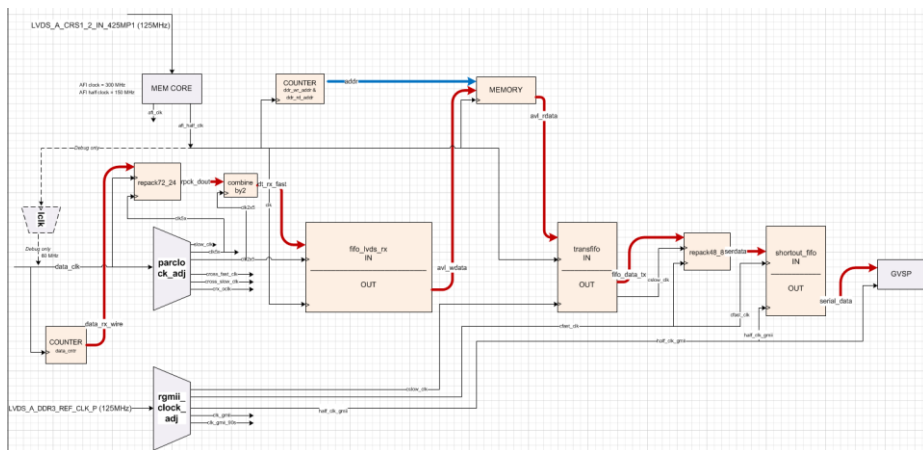
#### **3.1 Clock-domain synchronization**

Different modules and interfaces has different clock frequencies. So clock domains synchronization routine implementation was necessary. First In First Out (FIFO) blocks were used for this purpose. In Figure 1 structure of data storage, transformation and transmission modules for FPGA based set up is shown. There are two independent clock sources, one of which follows sensor-emitted data, the second – is high speed external memory driven base clock.

Every module is driven by finite state machine (FSM), which checks for data ready state, presence of data request and prevents output buffer underflow and overflow. Block diagram is shown in Figure 2.

#### **3.2 Data queue delay**

Different data transmission modules employed in design have different number of synchronization stages in queues. As a result, input FIFO data receiving from high speed external memory is based on a four-stage synchronization queue, but output FIFO data transmitting to Ethernet interface has a single stage queue. It influenced output data delay in a way it was eight clock cycles in first FIFO and two in the second. The problem in FPGA synthesis appeared because as soon as logic utilization exceeds 80% and some calculations are memory and logic intensive router faced at problem of too long interconnection wires. Because of it one of the signals will rise too late because of too long data path. This issue has to be taken into account during selection of an algorithm. Writing of additional synthesizer constraints is necessary if logic utilization is too high [3]. This constraints are described in Synopsys design constraints file, so that it can be converted easily into ASIC constraints file. It will help to optimize netlist structure and quality of routed topology [4].



**Fig.1.** Structure of data transmission modules.

### 3.3 Optimization, bit shift repairing, data recovery

Incoming data is synchronized to data valid signal which was fed from sensor. However so many transformations caused in unclear placement of start line and start frame pixels. Start of line and start of frame markers were added to fix this issue. It helped to restore image borders due to marker placement.

Input data was synchronized according to the clock phase. Dynamic phase alignment routine was implemented for this purpose. Algorithm was based on finding a template sequence in data stream and shifting the data clock in according to the template reference clock carrier's frequency and phase. It chooses clock phase restoring template sequence correctly and use it to shift data reference clock.

### 3.4 Paralleling

All FPGAs were based on partly the same firmware (depicted in Figure 1). Some differences were in pin assignments of chip to chip data exchange modules, because ones are master and the others are slave devices. Master devices had external interface module and transmitted data over Ethernet connection.

## 4 ASIC implementation

### 4.1 ASIC advantages

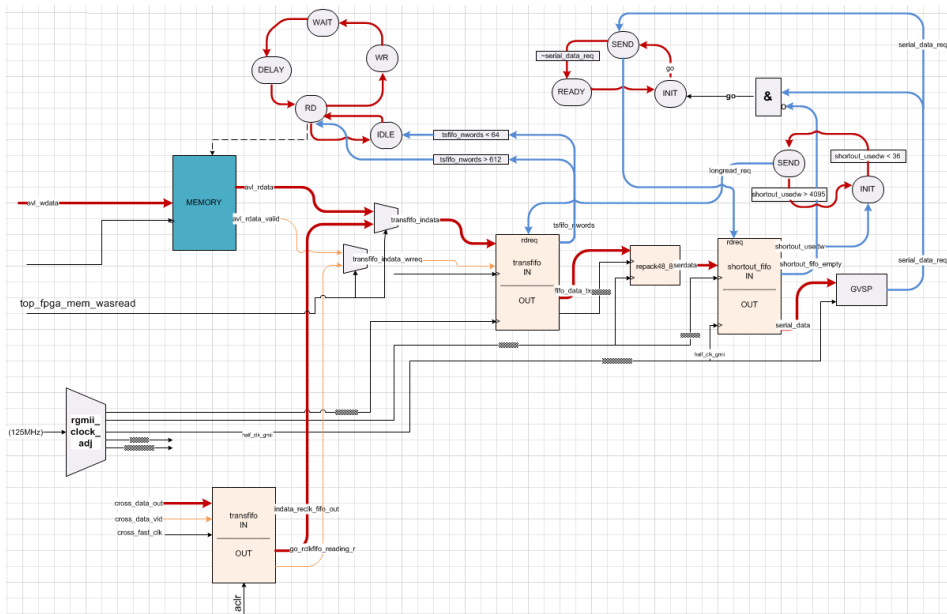
To exclude cyclic transmission of incoming data through multiplier queue and to reduce FIFO number and bus width transformation architecture was specially adjusted for ASIC implementation [5, 6].

After adjustment has been finished RTL code was compiled into logic primitives (netlist) and routed in silicon CMOS-structure.

### 4.2 Architecture

ASIC can not be reconfigured flexible against FPGA so additional mechanisms of function adjustments should be implemented. In according to this purpose control and status

registers were employed. Moreover to set values into registers and control data to memory and Ethernet transactions RISC-V CPU-core was implemented. CPU based on Syntacore SCR1 realization written in System Verilog. Data was transmitted internally over 32-bit AXI4-bus. It provided data control logic and it manages access to data bus, check whether it is busy or not. Ethernet medium access controller (MAC) was implemented and routed over AXI4 bus to CPU core. GigE Vision interface core was implemented in Verilog and connected to logic unit and external interface so that it works like intermediate chained routine. See Figure 3 for precise architecture description.



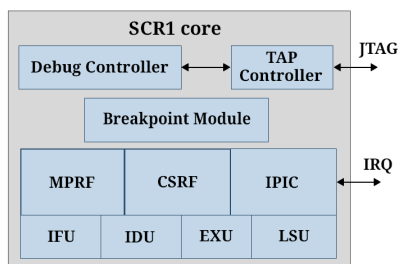
**Fig. 2.** Finite state machine placement and its role.

In result ASIC implementation can exclude FIFOs and difficult cyclic calculation due to increased buses width and multiplier queues. Also operating frequency will be increased and the demands on technological process precision will be considerably reduced.

### 4.3 ASIC netlist implementation

During FPGA debugging synthesis constraints were adjusted initially and it was the starting point to adjust ASIC synthesis constraints, which described timing and mapping restrictions. Also drive strengths and slew-rate settings to top-level ports of the design were determined. Timing-driven I/O register mapping was performed. Number of nodes driven by top level port instance was under control – fan-outs were restricted. This values were written in Synopsys Design Constraints (SDC) format. During RTL synthesis behavioral RTL code was transformed into structural RTL code according to technological cell libraries. In result behavioral Verilog file was converted into logic level Verilog implementation[8-11].

Additional netlist synthesis tools (RTL schematic, technology-mapped schematic, and critical path schematic) helped to make further constraints to optimize the design. In result technology-mapped netlist was generated.



**Fig. 3.** CPU modules structure.

## 5 Conclusion

Signal analysis algorithms developed for high speed camera were tested in FPGA and are ready to be implemented in silicon. It helped to perform image transformation in real time and decreased overall power consumption and data transmission on request delay during starting of device and after exposition is finished. Some data path bottlenecks were removed in according to parallel architecture optimization technique. In result power consumption was reduced and logic stability was improved[7, 12-13]. This factors helped to use device in different areas such as propulsion engine testing [14], aircrafts, observation balloons and space satellites.

## References

1. I. A. Kozulin, O. V. Nikanorov, S. N. Koreshev, J. Optical Tech **75(9)**, 558 (2008)
2. A. N. Samsonov, A Pat. Rec. and Im. Analysis: A. M. T. and A. **25(2)**, 255 (2015)
3. A. N. Samsonov, F. A. Bykovskiy, *Int. Conf. on M/Nanotech. and El. D.*, 160 (2011)
4. T. V. Perevalov, D. R. Islamov, A. A. Saraev, Tech. Phys. Letters **42(6)**, 601 (2016)
5. T. V. Perevalov, V. S. Aliev, V. A. Gritsenko, A. A. Saraev, V. V. Kaichev, Microelectronic Engineering **109**, 21 (2013)
6. D. R. Islamov , V. A. Gritsenko, T. V. Perevalov, O. M. Orlov, G. Y. Krasnikov, APL **109(5)**, 052901 (2016)
7. D. I. Rogilo, N. E. Rybin, L. I. Fedina, A. V. Latyshev, Optoelectr. Instr. D. Process. **52(5)**, 501 (2016)
8. A. G. Pogosov, M. V. Budantsev, E. Yu. Zhdanov, D. A. Pokhabov, A. K. Bakarov, A. I. Toropov, APL **100**, 181902 (2012)
9. K. V. Feklistov, A. G. Cherkov, V. P. Popov, Solid State Com. **242**, 41 (2016)
10. A. A. Lyamkina, D. V. Dmitriev, A. I. Toropov, S. P. Moshchenko, APL **110(1)**, 011103 (2017)
11. A. Regler, K. Schraml, M. Spiegl, K. Müller, J.J. Finley, M. Kaniber, J. Vuckovic, A.A. Lyamkina, J. Nanophotonics **10(3)**, 033509 (2016)
12. D. I. Rogilo, L. I. Fedina, S. S. Kosolobov, A. V. Latyshev, B. S. Ranguelov, Phys. Review L. **111(3)**, 036105 (2013)
13. D. I. Rogilo, L. I. Fedina, S. S. Kosolobov, A. V. Latyshev, B. S. Ranguelov J. Crystal Growth **457**, 188 (2017)
14. Fedor A. Bykovskii, Sergey A. Zhdan, Evgenii F. Vedernikov, J. Prop. and Power **22(6)**, 1204 (2006)