Report No 2003/E

CMOS optical sensor and readout electronics for LumiCal Alignment System

J. Błocki, W. Daniluk, M. Gil, E. Kielar, A. Moszczyński, K. Oliwa,
B. Pawlik, W. Słomiński[1], L. Suszycki[2], W. Wierba, L. Zawiejski

Abstract

The silicon-tungsten calorimeter LumiCal, located in very forward region of the future detector at the International Linear Collider, is proposed for precisely luminosity measurement. One of the requirements to fulfil this task is available information on the actual position of the calorimeter relative to the beam interaction area which should be known with accuracy of a few micrometers. In this paper we present project measurement unit for the positioning of the LumiCal electron detector by optical method using a laser beam and a CMOS sensor.

[1] Jagiellonian University, Cracow
[2] AGH, University of Science and Technology, Cracow

Introduction

In the future detector for the International Linear Collider (ILC, with colliding beams of electrons and positrons e^+e^-) [1], the very forward region is a particularly challenging area for instrumentation. The LumiCal detector [2] is expected to give a required precision luminosity measurement and to extend calorimetric coverage of small angles of electron emission from 28 to 90 mrad. The luminosity measurement will be based on detection of Bhabha event rate and a relative precision of the integrated luminosity of 10^{-4} will be enable. A precise measurement of the scattering polar angles requires an ultimate precision in detector mechanical construction and metrology. The crucial point is to monitor on-line the detector displacement under operation with respect to the colliding beams.

Requirements

The luminosity measurement requires extremely precise alignment of the two LumiCal detectors each to other and very precise positioning with respect to the beam line and the interaction point. Monte Carlo simulations have shown [3] that the inner radius of sensors layers have to be known with the accuracy better than 4 μm, the distance between calorimeters along the beam axis must be known to a accuracy of 60 μm over the ~4.5 m distance and the transversal displacement (x, y) with respect to the beam is required to be known to 600 μm accuracy (100-200 μm optimal). Initial inner radius of the detector can be measured in the lab using optical methods and precision movable table with the cross check of interferometer. The beam pipe is proposed as a suitable reference for the distance along the beam and transversal displacement and can be precisely surveyed before installing under different conditions (i.e. temperature). The temperature and tension sensors should be installed on the beam pipe to control and correct the mechanical dimensions. The Beam Position Monitors are mounted at well known position inside the vacuum pipe also and that would allow determining the actual position of LumiCal with respect to the beam position.
The position monitoring of the detector should not interfere with the beam pipe, hence a non contact system is preferred. We have chosen an optical laser system with a CMOS matrix sensor to measure the transversal (x, y) and longitudinal (z) displacement of the LumiCal with respect to the beam pipe flange. The position sensors will be placed between the rear side of the detector and beam pipe flange. The radiation dose in that area seems to be small because of shielding, but we can use radiation hard CMOS matrix sensors. The use of a few position sensors per calorimeter would allow to determine also the angle between detector axis and beam direction and assures better reliability in case of position sensor failure.

Optical sensor and readout electronics

Choosing the optical sensor for laser measurement of LumiCal's detectors displacement, application of CCD and CMOS sensors was considered.
For using CMOS speaks:

- integrated with the sensor picture processor,
- higher radiation resistance,
- single supply.

After comparing the producer's datasheets it is proposed to use Kodak's 12bites 1/3" CMOS KAC9618 sensor (optionally KAC00400). The structure of KAC9618 construction is based on:

- made in CMOS technology monochrome 648*488 pixels at pitch size of 7,5 * 7,5 µm matrix, that gives 30 frames for second,
- 12bites analog-to-digital converter construction.

The construction is controlled by I^2C-bus.

Application for Kodak's 1/3" CMOS KAC00400 sensor and 12bites analog-to-digital converter construction are proposed as an optional sensor. In the Kodak's 1/3" CMOS KAC00400 sensor's structure is implemented a color CMOS 768*488 pixels at pitch size of 6,7 * 6,7 µm matrix– the matrix gives 30 frames for second.

The construction is controlled by I^2C-bus.

Fig.1 shows the head board with optical sensor.

The final choice of the sensor will be made after prototype tests, so the mechanic-electric solution is based on separable PCB.

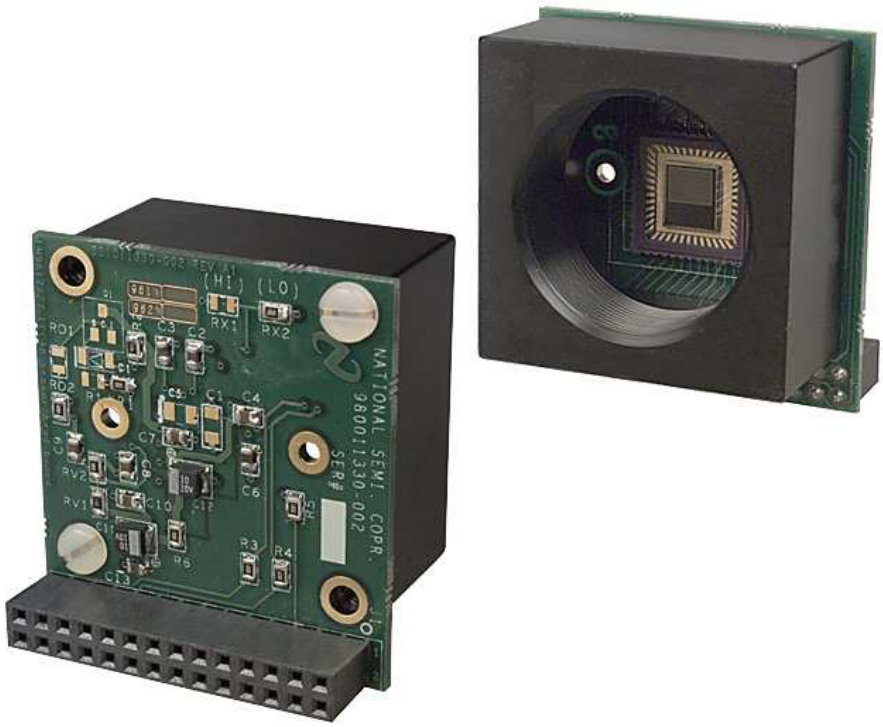

Fig.1. Head board with optical sensor

The XESS XSTend v.3.0 development board (Fig.2.) is used to prototype and test the design. The board is built based on a Xilinx Spartan-II XC3S1000 FPGA. The board also contains 32 MB (4M x 16bit x 4 Banks Synchronous DRAM LVTTL) of external RAM memory in the Samsung K4S561632A chip , as well as 512 KB (2M X 8 BIT / I M X 16 BIT) of flash memory. The on-board CPLD and parallel port interface makes it easy to load new designs into the FPGA. The board also provides several useful components for input/debugging such as a pushbutton, dip switches, etc.

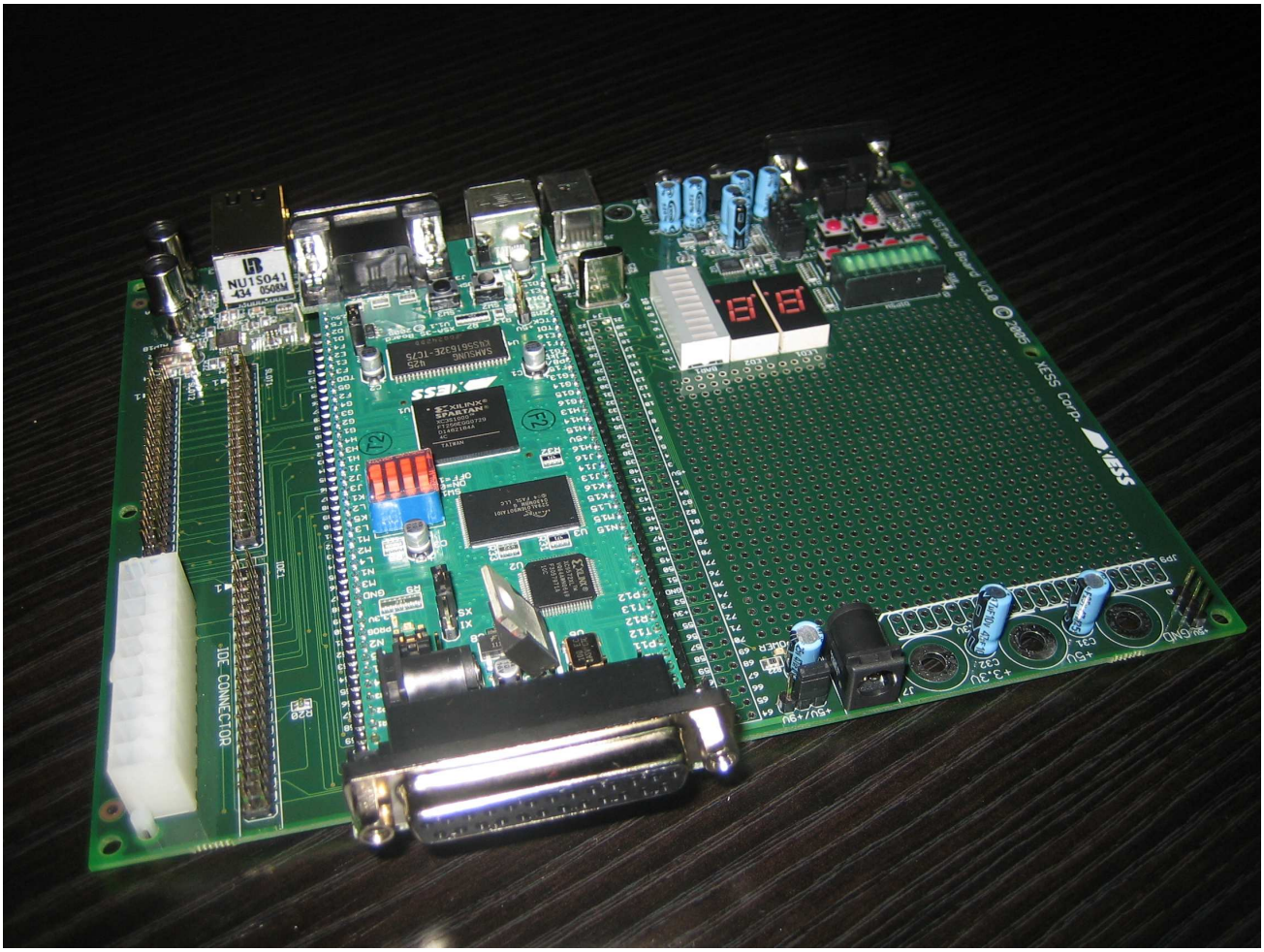

Fig.2. The XESS XSTend v.3.0 development board

The software placed in the FPGA is implemented in VHDL. VHDL provides strong type checking and allows us to build the design up from smaller modules. The program loaded onto the FPGA has three main tasks: to control the CMOS image sensor, to read and store pixel data from the CMOS image sensor, and to stream the pixel data to the host software through the parallel port. The software is divided into several modules, each performing a specific function. Block diagram of the data flow within the system is shown in Fig.3. (note: LM9618 = KAC9618).

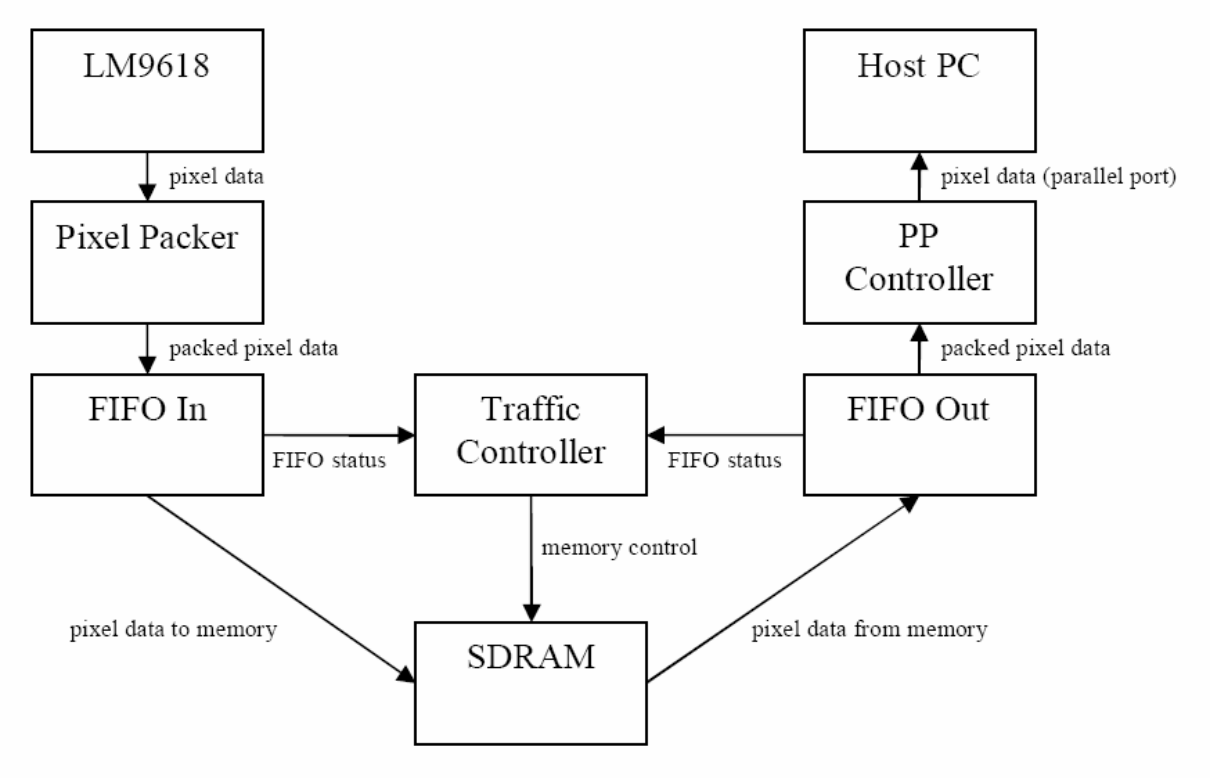

Fig.3. Block diagram of the data flow

I^{2}C Interface.

The I^{2}C interface is made up of two modules. The lowest-level module is the "simple_i2c" component. This component is a bare-bones I^{2}C interface available from http://www.opencores.com. Build on top of that module is the "lm9628_i2c_interface" module, which is based on a Dallas 1621 interface written by Richard Herveille, also available from opencores.com. The module provides an address, data, and several control lines for reading and writing values to registers via the I^{2}C bus. The top-level module (the *stream_camera_top* component) utilizes this interface in order to initialize the CMOS sensor with the correct settings in its control registers.

Storing Pixel Data.

The first step in storing images from the camera is to receive two eight bit pixels and store them in a single sixteen bit word. This is accomplished in the *pixel_packer* component. Once two pixels are received from the sensor, they are passed to a FIFO buffer in the form of a single word, with the first pixel received stored in the lower 8 bits and the second pixel in the upper 8 bits. The output data lines of the FIFO are connected to the input data lines of the memory interface, and the FIFO status signals are fed to the memory arbitration unit. The memory arbitration unit (the *traffic_controller* component) controls access to the memory interface.

Streaming Pixel Data.

The camera utilizes the parallel port to communicate with the host PC. Because of the way the interface CPLD is programmed on the XSA development board, there are seven

bits available for data from the PC to the FGPA (commands) , and four status bits available for data from the FPGA to the PC (image data). The *pp_interface* module controls the movement of data in and out of the parallel port at the lowest level. The lsb of the seven bit data bus is debounced by the *signal_debounce* component and used as the clock for the synchronous transfer of image data from the FPGA to the PC. Because only one nibble (4 bits) can be sent from the FPGA at a time, a second clock is generated that transitions with a rising edge every time the transfer of one sixteen bit word is complete. This allows the remaining layers of software to be unaware of the four bit limitation and transfer one full word at a time. The *pp_controller* module is one layer above the *pp_interface* module. Its main function is to read incoming commands from the PC and initiate frame transfers when appropriate. It provides a status signal to the memory arbitration unit as well to indicate when a frame is being transferred and pixel data from memory is needed.

Memory Arbitration / Control Logic.

Because the modules responsible for both outgoing and incoming data need access to the memory, some sort of arbitration is needed. The incoming and outgoing transfer rates are very different and not synchronized in any way, so in order to guarantee that we have a complete image to transfer at any given time, we need three separate image buffers. We need one buffer to hold the incoming pixel data, one buffer to hold the pixel data of the outgoing image, and one buffer to hold the last complete frame. These details are handled by the *traffic_controller* module. The module transfers data two pixels (one 16-bit word) at a time to both the input and output FIFO's. If no frame is currently being outputted through the parallel port and there is data to be read in the incoming FIFO, that data is read and stored in memory. If there is a frame currently being output through the parallel port, the module writes data to the outgoing FIFO until it asserts its "almost full" status flag. Because the speed of the memory is very fast in comparison to the speed at which data can be transferred over the parallel port, the system has no problem keeping up. Finally, if the outgoing FIFO is sufficiently full and the incoming FIFO contains new data, it is read and stored in memory. Each of the three frame buffers in memory has a unique address offset, making it simple for the *traffic_controller* module to switch between them. When a request for a new frame is received from the host, the offsets for the last complete frame and outgoing frame are swapped, so that the latest frame is transferred to the host. When the start of a new frame is received from the camera, the incoming and last complete frame offsets are swapped. On the rare occasion that the two events occur during the same clock cycle, the outgoing and incoming buffer offsets are swapped. Using this logic, we guarantee that incoming and outgoing images are never corrupted with data from other frames.

Acknowledgement

This work is partly supported by:
Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RII3-026126
and
Polish Ministry of Science and Higher Education, contract number 141/6.PR UE/2007/7.

References

KAC9618 datasheet.
KAC00400 datasheet
http://www.opencores.com
http://www.lems.brown.edu/~dec/docs