10.2478/v10048-008-0021-z
MEASUREMENT SCIENCE REVIEW, Volume 8, Section 3, No. 4, 2008

High Speed Optical Wavefront Sensing with Low Cost FPGAs

K. Kepa[1, 2 ,3], D. Coburn[1], J. C. Dainty[1], F. Morgan[2]

[1]Applied Optics Group, School of Physics, [2]Dept. of Electronic Engineering,
[1, 2]National University of Ireland, Galway, University Rd., Galway, Ireland
[3]Institute of Computer Engineering Control and Robotics, Wroclaw University of Technology, Poland
krzysztof.kepa@pwr.wroc.pl, {derek.coburn, c.dainty, fearghal.morgan}@nuigalway.ie

This paper outlines a study into deployment of a parallel processing scheme on an FPGA to meet the needs of high bandwidth processing in adaptive optics wavefront sensing. By exploiting a multi-stage pipeline approach we have significantly reduced the processing time needed to perform the wavefront sensing operation. The paper details the design, implementation, and performance testing results of the proposed FPGA-based wavefront sensing system.

Keywords: wavefront sensing, adaptive optics, FPGA, parallel processing

1. INTRODUCTION

ADAPTIVE OPTICS (AO) and the associated high-speed Wavefront Sensing (WFS) has seen much of its initial application as a tool for basic research in the field of astronomy (enabling the impact of atmospheric turbulence to be addressed in imaging the night sky). AO also has mainstream commercial applications in fields as diverse as optical communication, vision science, laser beam forming, microscopy [1], [2], etc.

An AO system senses the optical wavefront distortion (which degrades imaging system performance) and corrects it using active wavefront compensation. Fig.1 illustrates the AO system with wavefront sensor (used to measure image distortion), wavefront corrector (deformable mirror) and a high-speed control system, which changes the surface of the wavefront corrector to provide the necessary compensations.

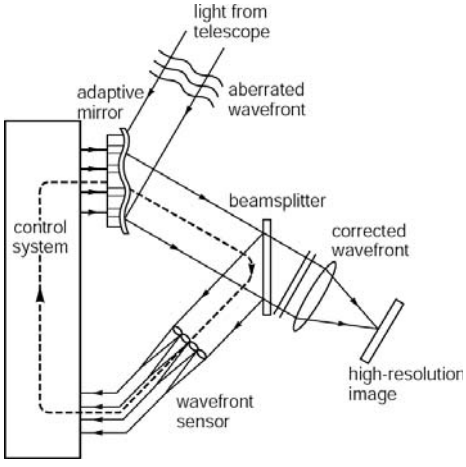

Fig.1 Adaptive Optics device.

The use of FPGAs in closed loop control systems for the next generation of high-end ground-based telescope AO controllers has been explored by a number of researchers [3, 4]. These studies look to address the AO system demands needed in realising the resolution capabilities of the next generation of ground based telescopes (in the ongoing push for larger telescopes primary mirrors). With such systems, the wavefront sampling and required control channels density places a huge demand on the real-time processing requirements of the AO controller. The ability of FPGAs to perform massively parallel processing increases the obtainable control loop frequency and reduces the computational latency in the control system. These systems tend to target top end FPGA solutions, since the focus is for the most part on performance with cost issues being a secondary concern.

In this paper, we focus on the benefits of a parallel computational model in the implementation of a low-cost WFS; related efforts are reported by Saunter et al. [5], [6]. Our design study has targeted a price point of low-cost Spartan-3 FPGAs. Throughout the study, efforts have been made to design and implement the system with design reuse and processing bandwidth in mind. Central in our effort is maximising the range of image sensor geometries that can be employed. The WFS processing architecture proposed in this paper is capable of processing multiple Regions of Interest (ROIs) within every image data frame.

2. SUBJECT & METHODS

Many factors affect the performance of the AO systems. Key among these is the time delay between sensor acquisition and corrective control being issued by the AO system (upper time-line in Fig.2). A typical approach is to charge general purpose CPUs or DSPs with the task of performing these calculations. In the majority of schemes, processing of data is blocked until all WFS image data are stored in local memory, thus introducing implicit buffering time overhead. This exacts a significant penalty on the potential processing power of the system. Even carefully crafted software implementations struggle to meet the 1ms processing requirements needed to realise AO control at a 1kHz frame rate. The key design approach proposed in this paper performs parallel processing of the image pixel data on-the-fly, i.e. as it is obtained from the image sensor. Our study has focused on methods of deploying FPGA acceleration for use in the Shack-Hartmann (SH) WFS.

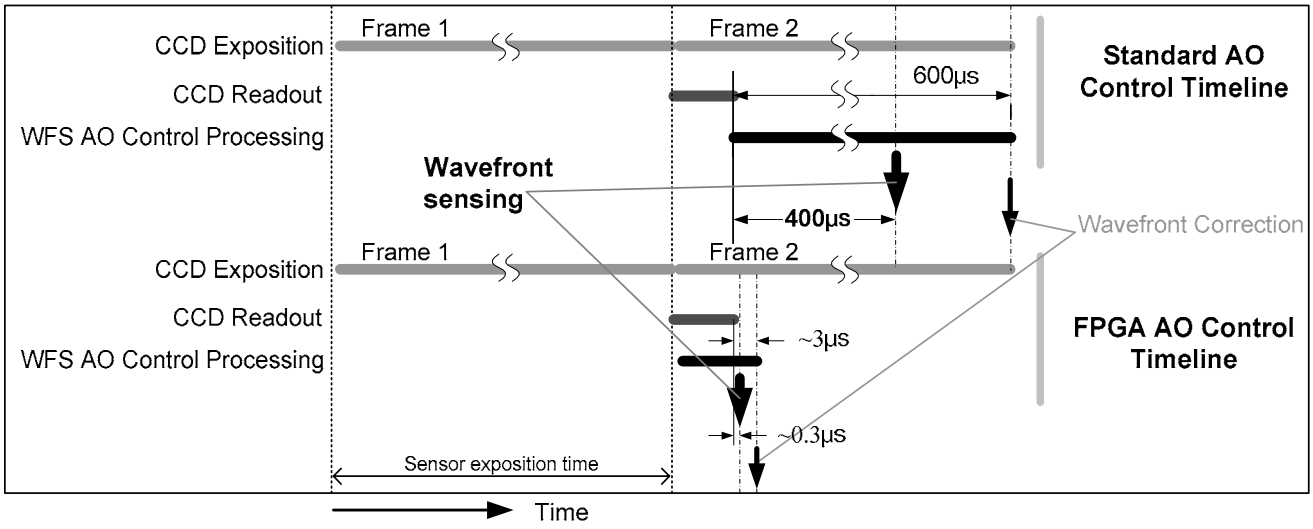

Fig.2 PC and FPGA time-line comparison for the optical communication system.

A. Low-cost FPGA devices

FPGAs offer an alternative to mask-programmed Application Specific Integrated Circuits (ASICs) and standard-cell Application-Specific-Standard Products (ASSPs) for mid- and low-volume systems. FPGA-based designs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASIC development. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

Low-cost FPGA device families, such as Spartan [7], Cyclone [8], are specifically designed to meet the needs of high volume, cost-sensitive electronic applications, such as consumer products. These cost-reduced FPGAs use simplified architectures, though follow high-end FPGA devices, without including many of their advanced features. Current low-cost FPGAs are now capable not only to accommodate loosely coupled functions, but can contain a complete System-on-a-Programmable-Chip (SoPC), e.g. Microblaze [9] or NIOS-II [10].

Our design study focuses on the use of a Spartan 3/3E target platform, together with associated design tools [11], [12].

B. Shack-Hartman wavefront sensor

A Shack-Hartman (SH) WFS assesses the wavefront by sampling a series of discrete subapertures across the wavefront [13], [14]. In the case of atmospheric turbulence imaging, the incident wavefront is distorted as it travels through the turbulence domain (Fig.3). In the standard configuration the Lenslet Array (LA) images portions of the aberrated wavefront on to an Image Sensor Array (ISA - a CCD or CMOS sensor placed in the focal plane of the LA) to form a series of spots, each spot corresponding to a different lenslet (or subaperture) in the LA. The position of these spots, for the incoming wavefront, relays information about the local tip and tilt of the wavefront, allowing the aberration of the incident wavefront to be assessed. In the absence of turbulence, the undistorted wavefront forms a regular grid of diffraction limited spots located in the centre of subapertures on the ISA. In the presence of turbulence, spots become displaced, typically by a sub-pixel amount (see Fig.3). The key function of the SH WFS is to determine this spot motion either to measure the absolute aberration (for a wavefront measurement device) or to estimate the residual wavefront error (for a closed loop AO system). To obtain representative numerical values of the wavefront distortion (slope coordinates) a centroid position is calculated for spots in every subaperture within the WFS ROI.

In AO WFS, the required image sensor data bandwidth may peak at the order of 0.9 Gb/s (typical resolution 10 bits/pixel, 300x300 pixels/frame, 1k frames/sec). However, the core data of interest in the system (i.e. calculated centroid positions) may be many orders of magnitude lower, e.g. 0.3Mb/s (10-bit x- and y-error values per subaperture, 15 subapertures per frame, 1k frames/sec). The ability to offload the computationally intensive tasks to an FPGA processing unit offers significant benefits in freeing up the host PC resources for other application-specific tasks. FPGA implementation also provides a real-time, low-cost platform.

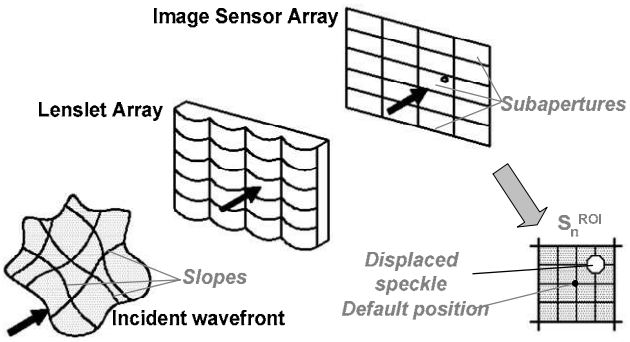

Fig.3 Shack Hartman wavefront sensor.

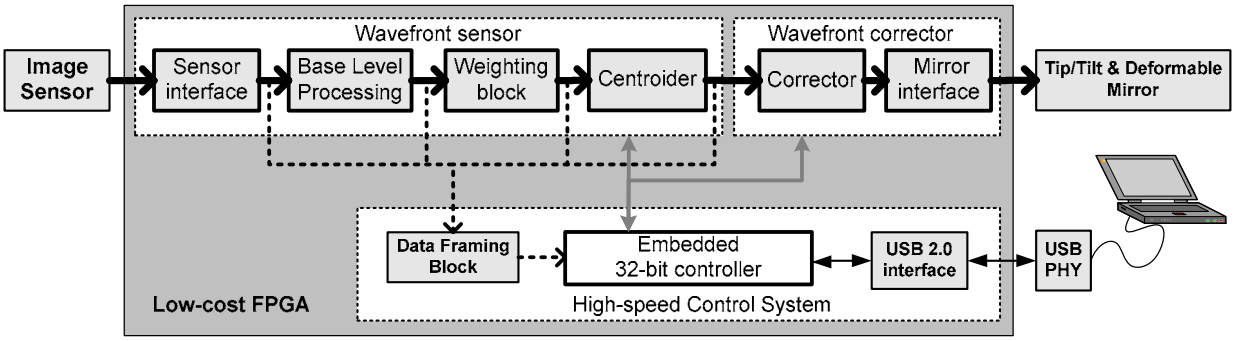

Fig.4 Low-cost adaptive optics control system architecture.

C. Wavefront sensor internal architecture

In this section the proposed WFS architecture is highlighted. The prototype platform under development supports a rich set of base level and SH slope processing algorithms. This section describes the architecture of elements within processing pipeline in Fig.4.

Central to this approach is the need to continuously maintain a free flow of data through the pipeline. Stalling of the processing by any pipeline stage impacts directly on the achieved bandwidth and buffering requirements possible with the proposed system.

1) Sensor Interface

The main task of the Sensor Interface component is to communicate with the external capturing device (image sensor), and to orchestrate pixel data transport from this device to the internal High Speed Data Path (HSDP) interface and onto the processing elements.

To cater for a range of image sensors with differing data transfer rates, clock-crossing synchronisation between the pixel clock and WFS internal data path clock is required (see Fig.5a).

To simplify the design of modules connected to the HSDP additional interface signals are generated within the Pixel Con-text Generator (PCG), encapsulating the Pixel Address (PixAddr), Frame Number (FN), the Real Time Clock (RTC), etc. The FN and PixAddr generated in the PCG provide information about the pixel-data position within the sequence of frames or within the frame itself. The RTC signal provides a timing signal obtained from a pre-scaled system clock. RTC timestamps are helpful in performance tests of the WFS and also in data post-processing and storage.

In the proposed WFS architecture, the image sensor configuration interface is implemented within the sensor interface block. This approach allows us to provide a unified programming model to a range of external devices and requires modification only to separate sub-designs to accommodate new communication protocols.

2) Base Processing Block

Most low-cost commercial CMOS image sensors suffer from significant levels of fixed pattern noise, which can be as large as 10% of the working range. Compensation is possible using base level processing which performs dark-image and flat-field correction on the image data [13]. In the design correction values are obtained during device calibration and stored in the external memory as a reference image (see Fig.5b).

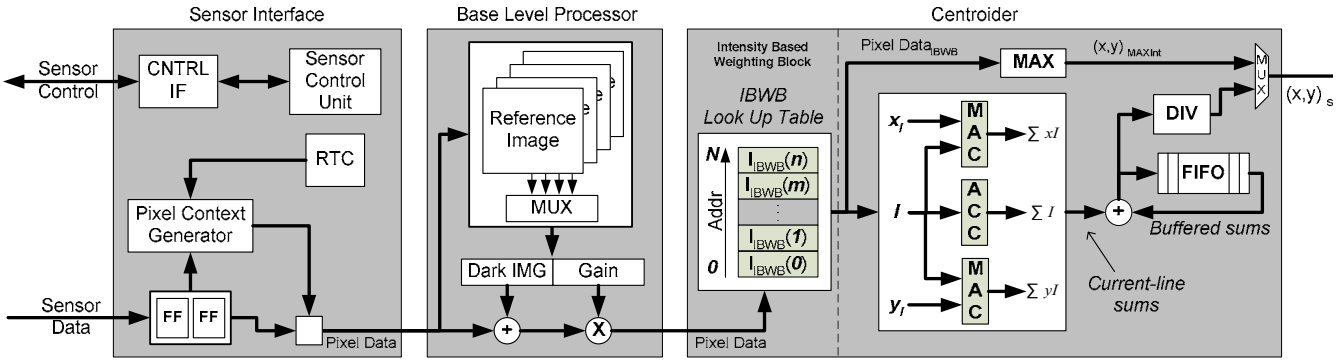

Fig.5 Wavefront sensor architecture: a) sensor interface, b) base level processor, c) centroider.

3) Centroiding Block

As detailed earlier, the image data received from the SH sensor represents the wavefront distortion across the set of subapertures, S^{ROI}. To obtain local metrics of the distortion $(x_n^{ROI}, y_n^{ROI})_{err}$, within a particular subaperture, the spots position has to be measured and compared against the spots reference position $(x_n^{ROI}, y_n^{ROI})_{ref}$, obtained during the calibration process (1).

$$x_{err} = x_{ref} - x_{spot}, \quad y_{err} = y_{ref} - y_{spot} \tag{1}$$

$$x_{spot} = \frac{\sum_i x_i I_{i,j}}{\sum_i I_{i,j}}, \quad y_{spot} = \frac{\sum_j y_j I_{i,j}}{\sum_i I_{i,j}} \tag{2}$$

Following the standard estimation approach, a calculation of the centre of the spot (centre of mass-like calculation) is performed following (2).

According to (2), all pixels internal to S_n^{ROI} need to be processed to obtain $(x_n^{ROI}, y_n^{ROI})_{spot}$, representing the spots position within subaperture. This enforces a requirement for data buffering since multiple subapertures can co-exist within a data line of a particular ROI (see Fig.6).

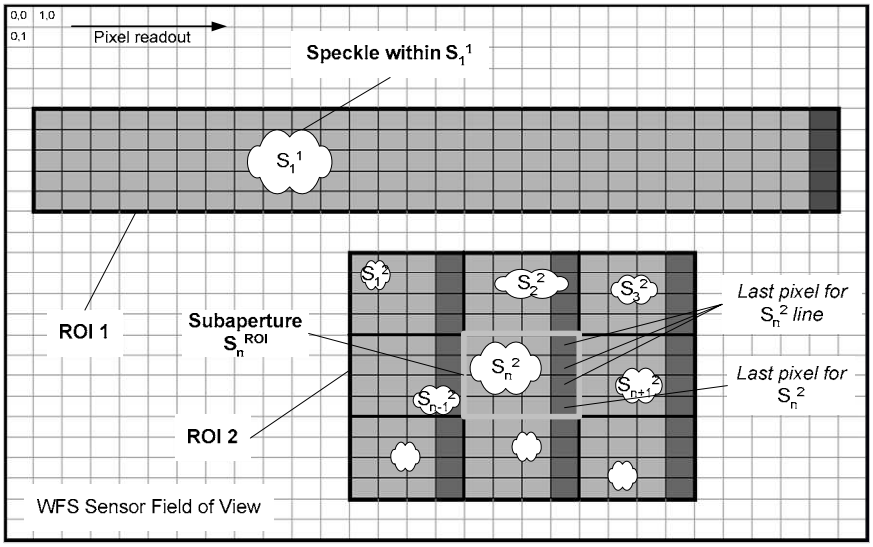

Fig.6 Test image with two ROI's: a) Single-subaperture ROI, b) 9-subaperture ROI.

In our design, pixel data is processed as it flows through the centroider block (see Fig.5). At the end of the subaperture region (Fig.6), accumulated (partial) sums from (2), namely $\Sigma_i x_i I_{i,j}$, $\Sigma_j y_j I_{i,j}$, and $\Sigma_I I_{i,j}$, are stored in a FIFO buffer (the order of data is maintained) and processing of the next subaperture S_{n+1} begins. On the next line, when S_n is processed, previously stored sums are retrieved from the buffer and merged with currently accumulated values before sending back to the buffer again. This algorithm repeats for all data lines within S_n^{ROI} subaperture. When the last line of the subaperture set is processed, the merged sums form the final result (no consecutive store step required) and the division process can occur. This approach minimises data transfer and buffering overhead as only partial results are stored.

Fast division. Inherent in the centre of mass calculation (2) two integer divisions must be performed per subaperture. Initially, a fully pipelined division module was embedded in the design to address this need using the Xilinx CoreGen tool. This exposed significant resource requirements, i.e. over 6200 Flip Flops (FFs) and 2000 Look-Up Tables (LUTs) and calculated the quotient in more than 20 steps (clock cycles in, one bit/step and divisor greater than 20 bit). This resource hungry solution proved to be unacceptable for low-cost Spartan-3 devices. It also turned out to be significantly time inefficient since subapertures can be as small as 4x4 pixels in size (only 4 steps per MAC per image line).

The follow-on design exploits the fact that the quotient in (2) is constant and bounded by the subaperture size k and desired sub-pixel resolution j. Thus the division result is ready after $k+j$ steps, i.e. 10 clk cycles for subaperture size of 64 pixels and 1/16 pixel resolution. At the expense of doubling the internal structure size, we further slashed it by a factor of two (down to 5 clk cycles) by calculating two bits of the quotient (instead of one) in every clock cycle. The resulting core used 188 FFs and 338 LUTs.

The Intensity-Based Weighting Block (IBWB). In a SH WFS, a range of optimal estimators exists for centroid calculation depending on the light level and noise characteristics of the image sensor [15]. Estimators may be implemented in the system by enabling weighted and constrained centroid calculation, as appropriate. This requires additional processing to be applied to the pixel data before centroid position is calculated.

The IBWB supports a wide variation of possible contrast enhancements, i.e. $I_{IBWB} = I_{IN}^{1.5}$. In the general approach, the IBWB transfer function can support a range of linear and non-linear functions applied to the input data I_{IN}, where

$$I_{IBWB} = f(I_{IN}) \quad (3)$$

The employed IBWB exploits the property of finite granularity of the pixel data. As resolution of the low-cost image sensors considered in our application is 10 bits per pixel, it offers only 2^{10} possible intensity values. Knowing that, we assumed the Look-Up Table (LUT) implementation of IBWB would be the most suitable approach (see Fig.5c). The input n-bit intensity value selects (addresses) the output m-bit word from the LUT. Thus, the required bit size of a LUT can be calculated as

$$LUT_{SIZE} = 2^{n+m} \quad (4)$$

Available EDA tools for FPGA design, e.g. Xilinx ISE CoreGen [ISE], provide various solutions to implement LUTs (memory elements) within the design, e.g. BRAM, distributed memory, LUT-RAM. Selection is dictated by the amount of the memory, and required performance. After review, the BRAM primitives were decided upon as the optimum option, as they provide high capacity storage and high bandwidth. Also, they do not occupy user logic resources.

4) Data framing block

For test purposes, a data framing block is included at the end of the pipeline to enable data to be read back by an embedded processor. The memory element is BRAM, generated using the CoreGen tool, and extended with a simple pixel counter and OPB interface.

5) Embedded controller block

The Embedded Controller Block (ECB) is mainly used to setup and control data path blocks. It also provides high-speed (USB 2.0, Ethernet) communication links to external devices, i.e. PC or high-resolution scientific camera.

D. Test Pattern Generator

Validation of the system performance is performed using a purpose built Test Pattern Generator (TPG) developed as part of this work. The TPG is a standalone System-on-Programmable-Chip (SoPC) design (Fig.7). Its flexible design emulates a sensor device by providing compatible Bus Functional Models (BFMs) and interfaces. Using a series of software scripts run on a PC, the TPG was propagated with data to provide real time WFS test datasets to verify system operation and performance. In the test setup TPG is used as an external device connected in place of the image sensor. Since the external TPG generated signal integrity problems (alignment with clock edge, etc.) at higher clock rates (above 40MHz), implementing the TPG core as a WFS internal module (within the same FPGA as the pipeline) would offer an advantage. The advantage is that this would support calibration and provide Built In Self Test (BIST) functionality to the design.

3. RESULTS

E. Performance

For preliminary testing purposes, the wavefront sensor prototype was implemented on a Virtex4LX-25 development board without any architecture-specific optimisations (limiting resources usage to that available to lower-end Spartan device). Initial tests demonstrate a maximum processing speed of ~83MHz. This translates directly to a sustained performance of 83 Mpixels/s, supporting the realisation of quite dense spatial WFS, e.g. 400 subapertures (16x16 pix) with a frame rate of 810 frames/s. It should be noted that the achieved performance can be extended beyond the 10 bit dynamic range of the analogue to digital sampling for the image sensor considered in the study. Increasing the dynamic range has a direct, near-linear impact on FPGA resource usage. However, the performance metric remains mostly unaffected, due to the use of a pipelined architecture. Only a few extra processing cycles

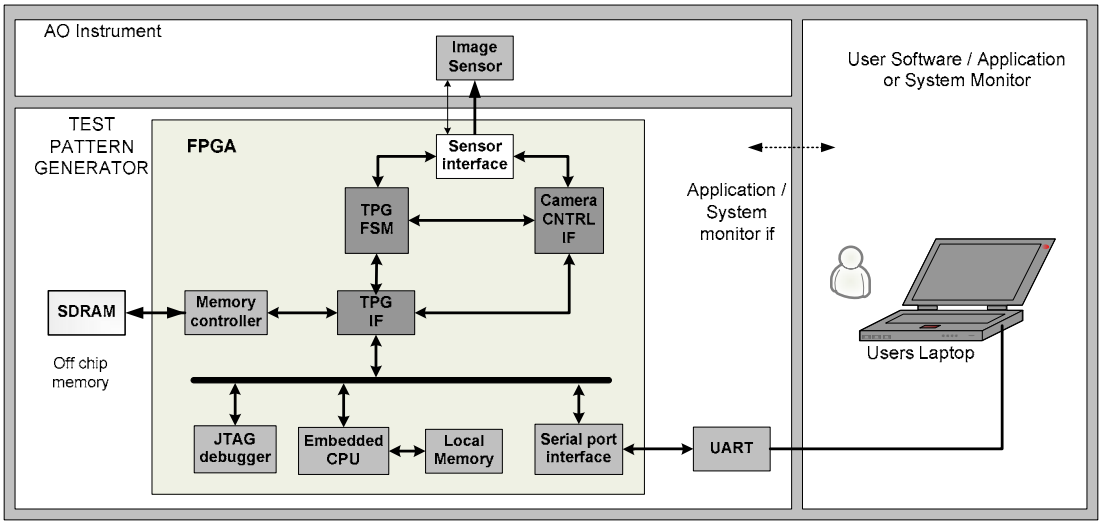

Fig.7 Architecture of Test Pattern Generator System-on-a-Programmable-Chip

may be introduced, impacting on process timing at the sub-microsecond level.

All processing within the design is performed using signed integer format enabling the SH spot locations to be calculated to 1/16 (or 1/32) of a pixel precision (typical of the level needed). Higher precision is possible with minimal impact on processing delay. The design incorporates a hand crafted divider unit which enables the centroid division to be performed within 5 clock cycles (at 83MHz). The maximum processing speed is determined by a long propagation chain within the hardware divider unit caused by the generic structure of the internal multi-bit adders and comparators. However, when compared to the CoreGen divider core, 80% of the logic resources are saved. The performance achieved supports all possible operation cases considered to date, as a pixel data rate greater than 80MHz occurs only in applications where a high-speed CMOS image sensor is required.

F. Functionality

As described earlier this design has been implemented with design reuse in mind. In practice, it can be customised using appropriate control and functional modules to meet specific designer needs, i.e. processing 15 subapertures only.

In its current realisation, the system is capable of supporting up to 400 subapertures of mixed size, ranging from a quad cell to 64x64 pixel subaperture geometries. In addition, the system can support multiple WFS pupils with up to 7 distinct sensing areas from a single image sensing device. All aforementioned features can be enabled on the pixel stream to produce slope estimates within ~0.3μs (exact number depends on design parameters set up in the initial build which may impact on process timing but at the sub-microsecond level) of the image sensor issuing the last pixel value.

The benefits of employing an FPGA are most directly visible when compared with the processing timelines of alternative approaches. As detailed, Fig.2 illustrates a PC and FPGA time-line comparison for the line-of-sight optical communication system. The PC solution is custom in-house software, being a considerable design effort in itself. The reduced processing delay of the FPGA system enables improved WFS performance and controller bandwidth. The main bottleneck is moved from data processing to the task of reading the data from the image sensor. Further improvement is clearly possible by employing multichannel readout from the image sensor, implemented in one instance by using multiple image sensors with appropriate optics.

G. Resources

Tab.1 details current FPGA resource usage. Further scaling of the WFS system to extend parallel stream processing with multiple instances of the core logic is quite feasible, even in low cost FPGA devices (see section 2.3).

WFS BLOCK	SIZE [SLICES, BRAMS]	SPEED [MHZ]
ONE PASS CENTROIDER	828 (7%), 1	80
ONE PASS CENTROIDER W/O DIVIDER BLOCKS	430 (5%), 1	157
FRAMING UNIT	23 (<1%), 20	335
BASE LEVEL (WEIGHTING UNIT)	10 (<1%), 1	N/A - under development

Tab.1 Wavefront sensor blocks resource usage and achieved performance for Virtex4LX-25.

4. DISCUSSION

In keeping with results of high-end FPGA-based WFS systems, this design study indicates a clear advantage of employing FPGA-based processing over the traditional CPU- or DSP-based approaches. The performance gain is mainly due to the removal of the so-called "von Neumann bottleneck" (limited throughput between memory and CPU), use of shadow buffers changed on-the-fly and the highly parallel, fine-grained FPGA architecture.

The presented WFS system prototype includes a sensor interface, pixel weighting unit (IBWB), centroider, and an embedded controller with USB 2.0 interface (Fig.4).

Design validation and system performance metric have been obtained using TPG. The CMOS sensor module is currently being tested with the system.

Current activity is focused on the implementation of a number of blocks for base-level processing, frame capturing and on further studies of feasible implementation of controller block and wavefront corrector.

5. CONCLUSIONS

This paper details the design, implementation, and performance testing results of the proposed FPGA-based wavefront sensing system. The low-cost FPGA-based multi-functional WFS block reported offers significant improvements (in terms of performance) for kHz image sampling. This performance improvement does not penalise dense spatial wavefront sensing. The best optimisation in bandwidth operation (minimal time between acquisition and correction) is achievable by implementation of the complete AO control system using FPGAs.

An added advantage of the design is the reduced power consumption (< 10W) over conventional PC-based designs. Power budgets are a prime concern to the large scale deployment of AO in fields such as optical free space communication.

Current work is focused on full system integration. Work is ongoing on the development of a scalable architecture of a MIMO block for use in either real time wavefront reconstruction or system control input. Parameterised and automated system configuration is also being investigated.

Future work will focus on the implementation of a second generation speculative centroider for constrained slope estimations and on addressing data transfer rate and scalability issues of the WFS. The long-term goal is the deployment of a complete low-cost AO control system, fully implemented in FPGA technology.

ACKNOWLEDGMENT

The project is funded by Enterprise Ireland grant no. PC/2005/060 and CFTD/06/317 and by Science Foundation Ireland under Grant No. SFI/07/IN.1/1906.

REFERENCES

[1] Greenway, A. and Burnett, J. (2004). *Industrial and Medical Applications of Adaptive Optics. Technology Tracking.* IOP Publishing Ltd.

[2] Ed. Dainty, C. (2007). *Adaptive Optics for Industry and Medicine.* Imperial College Press.

[3] Rodriguez-Ramos, L. F., et al. (2005). FPGA adaptive optics system test bench. *Proc. SPIE* 5903, 120-128.

[4] Goodsell, S. J., et al. (2005). FPGA developments for the SPARTA project. *Proc. SPIE* 5903, 136-147.

[5] Saunter, C. D., et al. (2005). FPGA technology for high speed, low cost adaptive optics. *Proc. SPIE* 6018, 429-435.

[6] Saunter C. D. and Love, G.D. (2007). Low-Cost, High-speed for Adaptive Optics Control. In C. Dainty (Ed.) *Adaptive Optics for Industry and Medicine*, Imperial College Press.

[7] Xilinx. (2008). *Spartan-3 FPGAs.* Retrieved April 20, 2008, from http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/index.htm

[8] Altera, (2008). *Cyclone III FPGAs.* Retrieved April 20, 2008, from http://www.altera.com/products/devices/cyclone3/cy3-index.jsp

[9] Xilinx. (2008). *Spartan-3 Evaluation Kit* Retrieved April 20, 2008, from http://www.xilinx.com/products/boards/s3_sk_promo.htm

[10] Altera, (2008). *NIOS II IDE* . Retrieved April 20, 2008, from http://www.altera.com/products/ip/processors/nios2/tools/ide/ni2-ide.html

[11] Xilinx. (2008). *ISE Foundation Software* . Retrieved April 20, 2008, from http://www.xilinx.com/ise/logic_design_prod/foundation.htm

[12] Xilinx. (2008). *Platform Studion and The Embedded Development Kit (EDK).* Retrieved April 20, 2008, from http://www.xilinx.com/ise/embedded_design_prod/platform_studio.htm

[13] Hardy, J. W. (1998). *Adaptive optics for Astronomical Telescopes*. New York, USA: Oxford University Press.

[14] Tyson, R. K. (1998). P*rinciples of Adaptive Optics.* New York, USA: Academic Press.

[15] Kasper, M., et al. (1999). Increasing the sensitivity of a Shack-Hartmann sensor, Proc. of the Canterbury Conference on Wavefront Sensing, Canterbury.