Performance of a modal zone wavefront recovery algorithm (Hudgin) implemented on FPGAs for its use in ELTs

José Javier Díaz-García, Instituto de Astrofísica de Canarias (Spain); Alberto Dávila-González, Univ. de La Laguna (Spain); Luis-Fernando Rodríguez-Ramos, Instituto de Astrofísica de Canarias (Spain); José-Manuel Rodríguez-Ramos, Univ. de La Laguna (Spain); Yolanda Martín-Hernando, Juan-José Piqueras-Meseguer, Instituto de Astrofísica de Canarias (Spain)

ABSTRACT

The use of AO in Extremely Large Telescopes, used to improve performances in smaller telescopes, becomes now mandatory to achieve diffraction limited images according to the large apertures. On the other hand, the new dimensions push the specifications of the AO systems to new frontiers where the order of magnitude in terms of computation power, time response and the required numbers of actuators impose new challenges to the technology. In some aspects implementation methods used in the past result no longer applicable. This paper examines the real dimension of the problem imposed by ELTs and shows the results obtained in the laboratory for a real modal wavefront recovery algorithm (Hudgin) implemented in FPGAs. Some approximations are studied and the performances in terms of configuration parameters are compared. Also a preferred configuration will be justified.

Keywords: AO System, ELTs, FPGAs, Hudgin

1. INTRODUCTION

Changes in the diffraction index of the media, due to variations in parameters affecting the atmosphere, produce wave front distortions. These aberrations affect the spatial resolution of the objects and disperse the photons in a bigger area on the detector making it less effective in the detection of faint object. Today, and next generation telescope developments, claim for the detection of faint objects while obtaining the best spatial resolution possible. If with a 10 m telescope, located in an observatory with optimum observing conditions, a 0.4 arcsec structure can be resolved, when this resolution is improved down to 0.04 arcsec, the resulting system performance is similar to what could be obtained in a 100 m telescope under the same atmospheric conditions with no wavefront correction. This gives an idea of the limitations introduced by the atmosphere and the importance of AO systems in the next generation of telescopes where diameters from 30 to 100 meters are the goal.

2. AO FOR ELTS

Today most observatories provide AO facilities to allow a better spatial resolution as an additional feature. For next generation of telescopes the huge diameters make it mandatory. The lost in spatial resolution, which apparently results in an effective reduction of the total size of the telescope, is unacceptable as every portion of the collecting area has to be effective for scientific and economic reasons. From now on the existence of AO system is a need of the telescope and all ongoing projects consider the development of this as part of the activities to be carried out to provide a successful facility.

The size of these new infrastructures poses additional requirements to the AO systems. There is a scale factor problem, now a larger area of the waveform has to be sampled and better sampling resolution is required, but also an increase in the system overall performance that results in faster response is mandatory. To adapt the existing hardware to achieve the new requirements is not straightforward. The increase in calculation power is not obviously achieved by multiplying the existing HW. Extrapolating the total volume and power consumption of existing systems this turns to be a real limitation that prompts for new solutions. High performance, compact in size, reliability and low power consumption are parameters that guide the design of the new generation of AO systems.

3. AO SYSTEM FUNCTIONAL DESCRIPTION

The objective of an AO system that we consider here is the correction of the wavefront that reaches the telescope with aberrations due to distortions introduced by the atmosphere along the light path. The AO system requires three functions: To detect the wavefront, determine what waveform should have reached the telescope in absence of a distorting media such as the atmosphere, and generate all the required steps to produce a correction to compensate for the aberrations. This divides the AO system in three main subsystems:

- The waveform sensor: Detects the light entering the telescope
- The waveform detection: Using the information obtained from the waveform sensor determines the local variations of the waveform
- The waveform recovery algorithm: Using the information provided by the local variations of the incoming waveform recovers the information of the waveform that should have reached the telescope in absence of distortion
- The calculation of the actuations required by the compensating component: Once the waveform information is known then the compensation parameters that need to be applied to the compensator are derived.
- An active component: Usually a deformable mirror that may adapt to the required shape to compensate from the aberration.

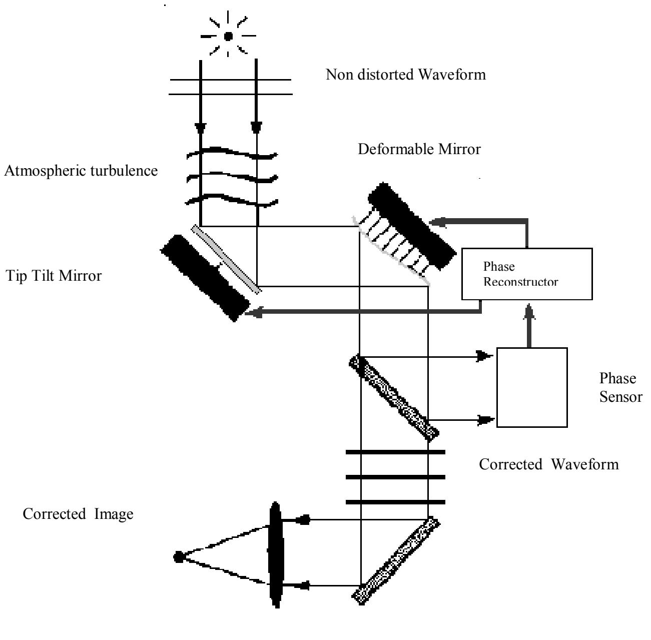


Fig. 1. AO System diagram

There is a time response required for the system. It is considered that the time variation of the atmosphere, for visible wavelengths, is in the order of 10 ms. If we want the system to be effective then the overall time response has to be less than this. It implies that the total time from the moment when the photons reach the detector of the wavefront sensor till the moment when the compensating mirror is in place has to be less that 10 ms. But a faster response is better as the atmosphere aberrations are not produced in steps but in the continuous.

In this paper an algorithm to recover the waveform, its implementation on hardware and final performances are described. Also, indications of key factors that would help to improve time latency for a complete AO system are mentioned.

4. AO SYSTEM HARDWARE

4.1 Shack-Hartmann sensor

4.1.1 Sensor description

A Shark-Harmann sensor is a well known waveform sensor that uses an array of microlenses in front of a detector to project subapertures of the waveform on it. The image of every single subaperture follow into a section of the detector and, in the case of absence of distortion, a single light point would be projected into the corresponding section of the detector. The effect of the aberration results in a displacement of such projection producing dx-dy displacements values for each subaperture.

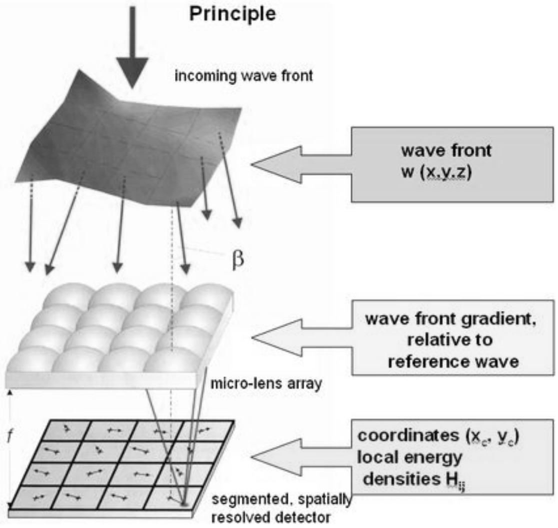


Fig. 2. Shack-Harmann sensor architecture.

The number of microlenses that compose the array determines the spatial sampling of the waveform. Every subaperture is projected into a section of the detector where the displacement has to be computed. The number of pixels corresponding to a subaperture determines the accuracy in resolving the displacement of the subaperture projection dx-dy.

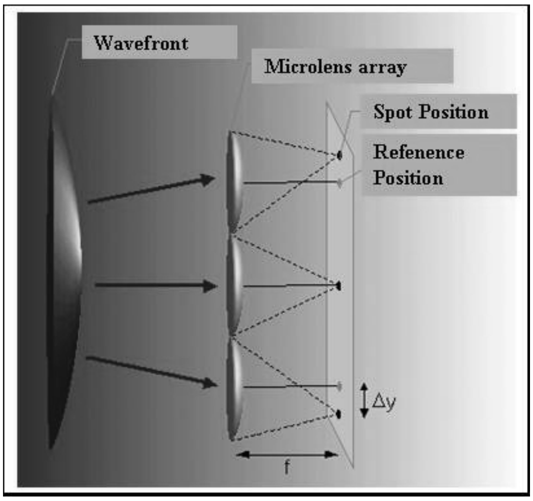


Fig. 3. Projection of subapertures onto the detector for a Shack-Hartmann sensor.

4.1.2 Estimation of the number of subapertures

For Extremely Large Telescopes (ELTs), with collecting apertures from 30 to 100 meters, the size of the focal plane is bigger than for present telescopes. Even if the spatial sampling resolution is to be maintained, the total number of subapertures will increase with a power of two factor. It is thus advisable to assume that the number of apertures required to obtain a reasonable spatial sampling is in the order of 32 * 32, which means that a 32 * 32 or larger microlenses array will be used.

4.1.3 Resolution. Number of pixels per subaperture

Also, to allow a better resolution in the gradient of the waveform in a certain subaperture, as a result of the atmospheric distortion and indicated by the displacement of the projection of every projection on the detector, the number of pixels per subaperture should be increased. To be on a safe side a number of 32 * 32 pixels per subaperture is considered.

4.2 DX-DY matrix calculation

After the detection of photons using the Shack-Hartmann sensor, the gradient of the waveform for each subaperture needs to be computed. Two matrixes DX and DY, with the dx and dy values corresponding to the deviations of the centroid from the nominal position for every subaperture, are obtained.

Different methods have been used:

4.2.1 Centroiding or Gravity method

This method uses the well known gravity or weight method where every pixel contributes with a factor corresponding to the total signal received multiplied by a factor representing the distance in x or y to the nominal position in the non distorted waveform.

4.2.2 Image Correlation

Also image correlation techniques are applied to each subaperture to compare consecutive images and to determine the DX and DY matrixes.

4.3 Waveform Recovery

Once the information of the local variations, gradients, of the waveform for every subaperture is known, then the next step is to obtain the value of the waveform starting from the information contained in the DX and DY matrixes. This method is the objective of this work and will be detailed.

4.4 Transformations and Deformable Mirror actuation

Once the waveform received is known, then the correction that needs to be applied to obtain a non distorted waveform has to be calculated and the physical signals to be applied to the correcting element, normally a deformable mirror, generated. This is the last process in the sequence that has to be repeated with a time response fast enough to produce the correction before the distortion of the incoming waveform changes significantly.

5. THE HUDGIN METHOD

5.1 Theory

Hudgin provides an iterative mathematical method that allows the reconstruction of a bidimensional function from the information contained in the local variations in both directions. The algorithm is represented for the following formula:

$$\phi^{(M)} = \frac{1}{4}(\phi_1^{(M-1)} + \phi_2^{(M-1)} + \phi_3^{(M-1)} + \phi_4^{(M-1)} + \Delta\phi_1 - \Delta\phi_2 - \Delta\phi_3 + \Delta\phi_4)$$

Fig. 4. Hudgin algorithm

M represents the algorithm iteration index, $\Phi1$, $\Phi2$, $\Phi3$ and $\Phi4$ represent the phase values of neighboring subapertures, in this case calculated in the previous iteration, and $\Delta\Phi1$, $\Delta\Phi2$, $\Delta\Phi3$ and $\Delta\Phi4$ the gradients.

The Hudgin method, as is based on local variations of the function, does not allow the recovery of the continuum value of the function. This is not necessary for our purpose as our need is to correct for the distortions.

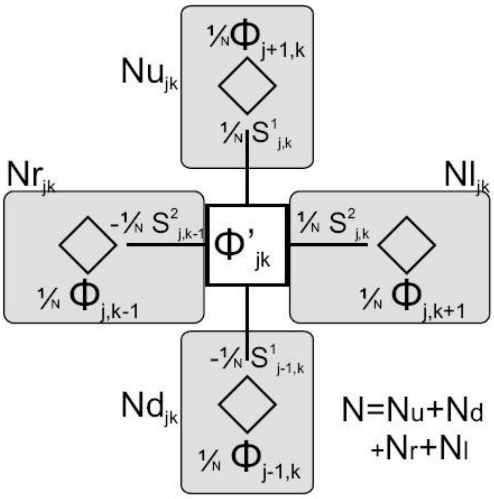

$\frac{1}{N}\Phi_{j+1,k}$
Nu_{jk}
$\frac{1}{N}S^1_{j,k}$
Nr_{jk}
Nl_{jk}
$-\frac{1}{N}S^2_{j,k-1}$
Φ'_{jk}
$\frac{1}{N}S^2_{j,k}$
$\frac{1}{N}\Phi_{j,k-1}$
$\frac{1}{N}\Phi_{j,k+1}$
$-\frac{1}{N}S^1_{j-1,k}$
Nd_{jk}
$\frac{1}{N}\Phi_{j-1,k}$
N=Nu+Nd
+Nr+Nl

Fig. 5. Phase and Gradients definition in a Shack-Hartmann sensor for the Hudgin method. u,d,r,l stand for up, down, right and left, j and k represent the file-column in the phase matrix and N represents the neighbor whose phase is at position jk in the phase matrix.

5.2 Numerical simulations

Some numerical simulations have been performed to understand the capabilities of this method and the feasibility to implement it in FPGAs where the floating point arithmetic is not efficient.

5.3 Accuracy vs number of iterations

The accuracy of the final result depends on the number of iterations performed. This directly affects the time required to obtain a result that matches, with reasonable error, to the actual waveform reaching the sensor. The actual number of iteration cycles may depend on the characteristics of the waveform function. It has been found that, with little difference, for Gaussian and Kolmogorov like waveforms iterations in the order of 500 give results with error results below 10^{-3} being the error in the order of 10^{-2} for about 200 iterations. More than 500 iterations will not produce significant error reduction and thus this figure can be considered as the required number of iterations for a complete recovery of the incoming waveform.

5.4 Singularities in the borders of the aperture

There is also a practical problem that has to be faced when applying the method to a limited area. Hudgin applies to infinite surfaces, and this is the case of light, but as our aperture is limited we have to work with limited waveforms to a certain area. We have to face the contour problems and the fact that there are missing neighbors in the borders of the aperture. It has been assumed that the missing subapertures required to complete the algorithm for the contour have the same value as the value of the aperture of interest. It has proven to be a convenient assumption as the numerical simulations provided good results

5.5 Numerical simulations

5.5.1 Simulations for a known function

For the shake of simplicity, to easy in comparing results, a Gaussian type signal was used to validate the inspection method consisting of:

- Generation of a Gaussian waveform

- Sample it into subapertures, and the values of the gradients dx and dy for each subaperture is calculated. At this point we had the DX and DY matrixes and the initial function that would provide such gradients. dx and dy values of missing subapertures in the contour are assumed to have the same value as the aperture of interest.

- Run the Hudgin method and analyze results comparing the output waveform with the initial Gaussian waveform in terms of the number of iterations.

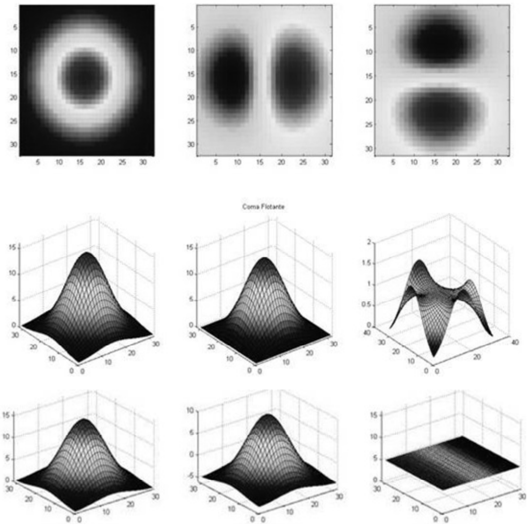


Fig. 1. From Top left to bottom right. Gaussian waveform, DX-DY matrices and comparison of original and recovered function with error for non border correction (middle) and with border correction (bottom)

5.5.2 Atmospheric like distorted waveforms

The actual application will deal with atmospheric like distorted waveforms. As to validate the method it is required to know the initial waveform to compare it with the result obtained after running the method then it is necessary to synthesize atmospheric like waveforms. Assuming that Kolmogorov statistics represents a valid model we used waveforms using this statistics, ran the Hudgin method and proceeded as with the Gaussian waveform. The result was that the method works well for this type of waveforms.

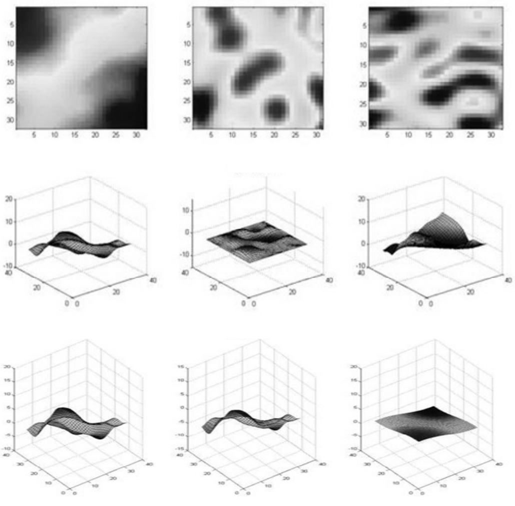


Fig. 2. Same as previous Fig but using a waveform complaint with the Kolmogorov statistics.

5.6 Algorithm implementation in FPGAs. Feasibility.

5.6.1 Fix ponint vs floating point arithmetic.

Once the validity of the algorithm to recover waveforms as those likely to be produced by the atmosphere has been proved, the next step is to provide a physical implementation of the algorithm into a HW able to perform the required calculations in the time frame imposed by the application. As mentioned before, the total time available to correct from waveform distortion, from the image acquisition till compensation, has to be less than 10 ms. It is then advisable to make this computation, and all the computations and actions required to compensate the aberrations, in the minimum time possible. This work pretends to produce dedicated HW to make this calculations at the fastest speed possible. The circuits will be implemented in FPGA.

The HW inside the FPGA is rather efficient if calculations are made in fixed point arithmetic. First it has to be demonstrated that the results, obtained with this arithmetic follows the results in the previous simulations with floating point arithmetic. A couple of assumptions have been made to provide relevant results:

- Number of subapertures: We assume that the maximum number of subapertures is 256 * 256. This parameter determines the maximum spatial information of the waveform. The spatial resolution.
- Number of pixels per subaperture: We assume that the maximum number of pixels corresponding to the area of the detector associated to each subaperture is 256 * 256. This determines the maximum resolution in the calculations of the dx and dy values.

We then require 16 bits where 8 bits determine the subaperture while the other 8 bits indicate the position of the centroid, dx or dy, within a particular subaperture.

These numbers are defined in excess and will give a worse case in terms of performance as the calculations will be more than those required by a real system.

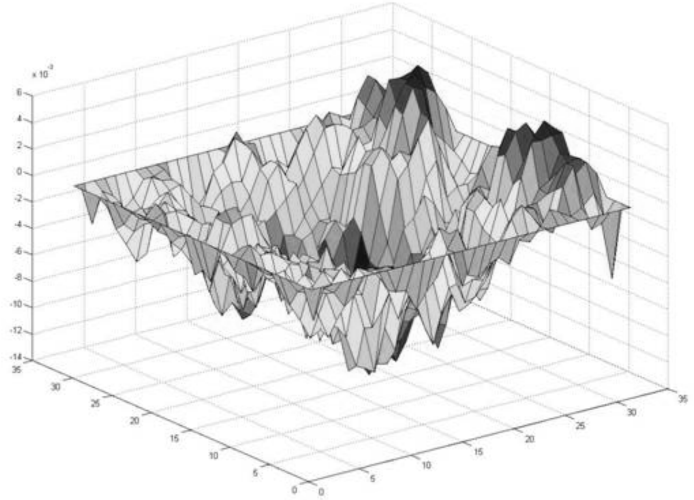


Fig. 3. Differences between Floating point and fixed point arithmetic simulations.

The results obtained using Fixed and Floating point arithmetic have been compared and the error found is in the order of 10-3. This is in the order of magnitude of the error in the waveform recovered for floating point arithmetic and thus it can be concluded that the use of fixed point arithmetic will not pose any significant error. The HW implementation on FPGA of the circuits to perform the Hudgin algorithm is therefore feasible.

5.6.2 Design architecture

The circuit architecture has been design to facilitate parallelism. To calculate the new phase matrix there will be as many circuits as files of subapertures operating in parallel. All the phases of a particular column of subapertures is calculated in parallel

5.6.3 Memory

The values corresponding to the initial differences dx and dy will be supplied in practice by a previous circuit that, using the centroiding, correlation or any other method will calculate them. As this is out of the scope of our circuit, these initial values are considered as available and stored in memory to be supplied to our circuit. There will be another memory, the phase memory, that is initially blank and will be updated with the phase recovered as a result of our algorithm. This memory is updated each iteration.

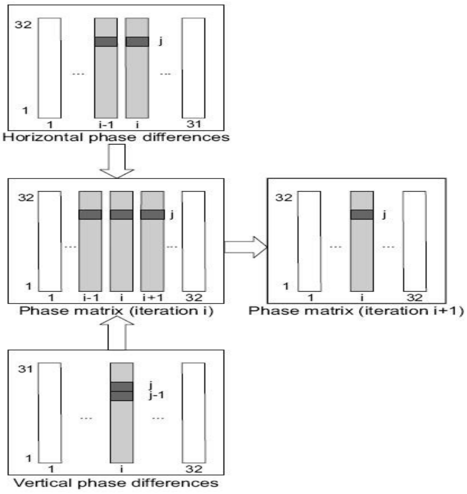

32
j
1
1
i-1
i
31
Horizontal phase differences
32
j
1
1
i-1
i
i+1
32
Phase matrix (iteration i)
32
j
1
1
i
32
Phase matrix (iteration i+1)
31
j
j-1
1
1
i
32
Vertical phase differences

Fig. 4. Memory architecture with DX-DY and Phase values.

5.6.4 State Machine

The process flow is controlled using a state machine that performs the same operations in every iteration cycle. The steps followed for every calculation consist of:

- Read the dx, dy and phase values: Required for the calculations of a particular subaperture. As the state machine is based on a column index step, a line phase calculator requires 1 phase value, 2 dx and 2 dy values. This means that the values of 3 consecutive columns are required to obtain the phase corresponding to the aperture under consideration. The result of the first column of phases is calculated after the data from the 2 first columns, plus the assumed values for column -1, are available.
- Operate the results: Once the dx, dy and phase values are available then the operations can be performed. There are a couple of singularities:
    - The subapertures in the borders do not have neighbours and the missing values for the phase are considered to have the same value as the phase to of the point under consideration during the previous iteration
    - There is no chance to make any operation till the values of the 2 first columns are received. Once the first 2 columns are read then the phase of the first ‘singular column’ is calculated assuming that there is a column 0 whose phase values equals the phase values of column 1.
    - Once this singular section has been covered then a new column of phases will be updated for every new column of subapertures that is read.

- To update the phase matrix: Once the new value of the phase has been calculated, before reading a new column of phases, the new values of the column are updated in the phase memory.

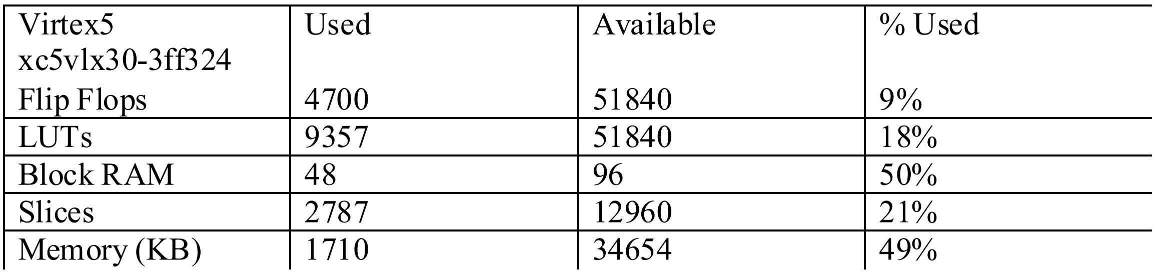
5.6.4.1 Regular and Singular apertures

It has been considered, for the sake of simplicity, that the aperture is a square. This implies that the apertures in the first and last columns, as those in the first and last rows of apertures are particular cases where their missing neighbors are substituted with the same phase value of the pixel under consideration. A distinction has been made in the algorithm to operate correctly in both cases named regular and singular apertures. It is to notice that there will be always singular subapertures defining the contour of the total aperture of the system.

5.6.5 Resources

The circuit has been described using VHDL as the behavioral description and planned for implementation on an FPGA. No assumptions of a particular FPGA architecture or custom resources have been taken into account and thus further optimization is possible by customizing the components according to the particular FPGA resources.

To provide the results and make an estimation of the benefits given by this hardware approach the circuit has been implemented into a VIRTEX 5 FPGA from Xilinx. It is considered for 32 * 32 subapertures and uses 16 bit resolution.



Virtex5 xc5vlx30-3ff324	Used	Available	% Used
Flip Flops	4700	51840	9%
LUTs	9357	51840	18%
Block RAM	48	96	50%
Slices	2787	12960	21%
Memory (KB)	1710	34654	49%

Fig. 5. FPGA resources in use.

This information includes also the memory block required to store DX and DY not necessary in a final implementation.

5.7 Accuracy results

The testing system was designed to allow introducing the same DX, DY matrixes as used in the mathematical computations. The phase matrix was recovered and stored in a file after the simulation of a circuit run. This resulting matrix was compared to the results obtained numerically. The same result was obtained from both methods for equal number of iterations and different input waveforms. This validates the calculations performed by the circuit inside the FPGA.

5.8 Time performance

A key parameter, that indicated the need of a HW implementation was the time required to run the algorithm. The system post place and route simulations were done and the results gave the following figure.

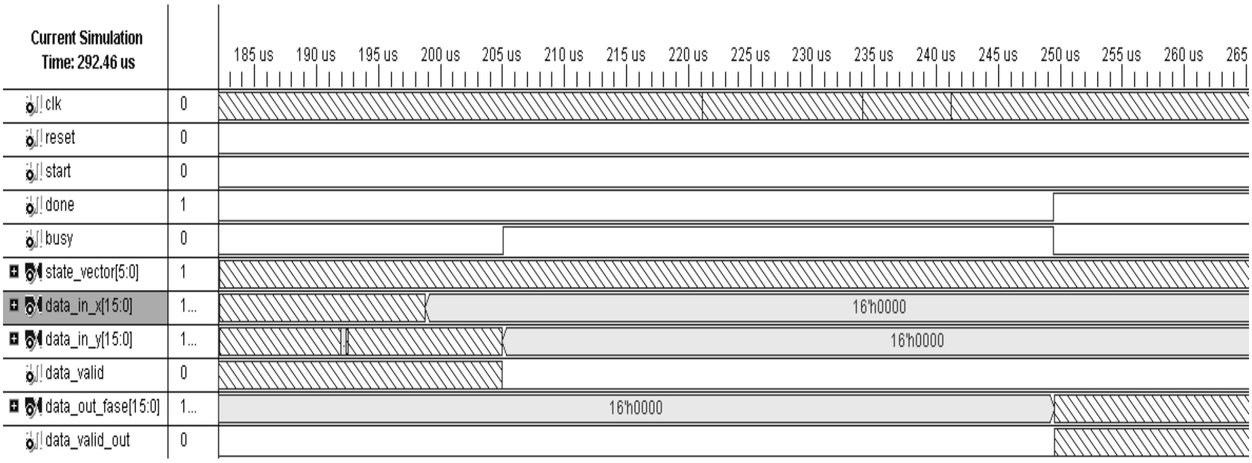


Fig. 6. Post Place and Route simulation results.

The result, obtained implementing the circuit in this particular FPGA but with no optimization to accommodate the circuits to the particular FPGA resources, for:

- with a 100 MHz master clock
- A full run with 10 iterations
- For 32 * 32 sub-apertures and 16 bit resolution

is 45 µsec which means roughly 4,5 µsec per iteration. If the error of the waveform recovered by the algorithm is acceptable after 200 iterations, then 0,9 ms are required to obtain the initial waveform.

This implementation makes it an ideal choice for AO systems such as those required by ELTs, where the number of phase sub-apertures 128*128 or even 256 *256 is considered.

- The time performance of this circuit is not a limitation factor in the AO system
- A proper reception sequence of the local variations of the wavefront allows processing in parallel while still receiving the remaining information.
- To allow a convenient sequence of local variation information the corresponding circuit and Phase sensor architecture need to be considered from the design phase.

5.9 Comparison with previous implementations

The fastest implementation known for this algorithm is based on GNUs and the performance of this implementation on FPGAs is about 2 orders of magnitude faster.

5.10 Optimization

The optimization effort to improve time performance and resource consumption has been limited and there is still margin to improve the results but with the present results it can be concluded that using the proper HW architecture of the AO system components a noticeable improvement in time performance could be obtained. Some of these key factors are mentioned.

5.10.1 Parallelism

It is advisable to run the algorithms in parallel and apply the corrections to different sections of the waveform as soon as they are available. The waveform variations take place along time and the fastesr the correction the better the result.

5.10.2 Detector Architecture

Referring only to the detector needs in terms of speed there is nothing new in this aspect. It is relevant that the processing of the DX-DY matrices can only been calculated once all the pixels corresponding to the subaperture are available. This indicates that the ideal detector architecture would provide an output channel per aperture and all this channels, al least those channels corresponding to subapertures in the same column of subapertures, are read and delivered with no delay. As this may imply too many outputs, the configurations should be done in such a way that the pixels corresponding to a same column are read at the same time with no dead times.

5.10.3 DX-DY HW Improvements

If the Hudgin algorithm is to be used then the subsystem used to calculate the difference matrixes should deliver the DX and DY values column after column to facilitate the process as planned. Every new column will provide enough information to start running the algorithm to recover the phase of every aperture in the column.

5.10.4 Waveform recovery

Finally the Hudgin method would run as described in this paper but iterating not the full matrix of apertures at a time but every column of subapertures independently. This would produce a column of recovered phases after the iterations of such column of subapertures is finished.

6. CONCLUSIONS

- The time performance of this circuit is not a limitation factor in the AO system
- A proper reception sequence of the local variations of the wavefront allows processing in parallel while still receiving the remaining information.
- To allow a convenient sequence of local variation information the corresponding circuit and Phase sensor architecture need to be considered from the design phase.

REFERENCES

[1] - José G. Marichal Hernández, Luis F. Rodríguez-Ramos, Fernando Rosa, José M. Rodríguez-Ramos. “Atmospheric wavefront phase recovery using specialized hardware: GPUs and FPGAs” (2005).

[2] - “Detección de frente de onda. Aplicación a técnicas de alta resolución espacial y alineamiento de superficies ópticas segmentadas”. José Manuel Rodríguez Ramos.

[3] - Yolanda Martín, Luis Fernando Rodríguez, Marcos Reyes. “Fixed-point vs Floating-point arithmetic comparison for adaptative optics real time control computation”

ACKNOWLEDGMENTS

This work has been partially funded by the Programa Nacional I+D+i (Project DPI 2006-07906) of the Ministerio de Educación y Ciencia of Spain, and by the European Regional Development Fund" (ERDF).