Real-time processing for the ATST AO system

Kit Richards
National Solar Observatory
Thomas Rimmele
National Solar Observatory

ABSTRACT

The real-time processing requirements for the four meter Advanced Technology Solar Telescope extended source high order adaptive optics system will be approximately 15 times that of the Dunn Solar Telescope AO systems on which the ATST AO system is based. The ATST AO, with its approximately 1232 subapertures, will use massively parallel processing and is based on Analog Devices TigerSHARC DSPs as the central processing units. We will discuss the requirements for processing and data handling and the architecture of the correlating Shack-Hartmannn and reconstructor processing unit and present the results of bench-mark testing of the DSP hardware that was selected for the ATST AO system.

1. INTRODUCTION

Atmospheric seeing has limited the resolution of larger solar telescopes until the adoption of adaptive optics (AO) such as the extended source Shack-Hartmann system developed by the National Solar Observatory and now in use at the 76cm Dunn Solar Telescope (DST) [1], Sacramento Peak, New Mexico. The success of this solar adaptive optics system is vital to the National Solar Observatory's (NSO) justification of building a four meter solar telescope, the Advanced Technology Solar Telescope (ATST) [2]. ATST's larger aperture combined with an adaptive optics system will provide unprecedented resolution for solar observation. The ATST is nearing completion [3] of its design phase and its intended site is on Maui's Haleakalā.

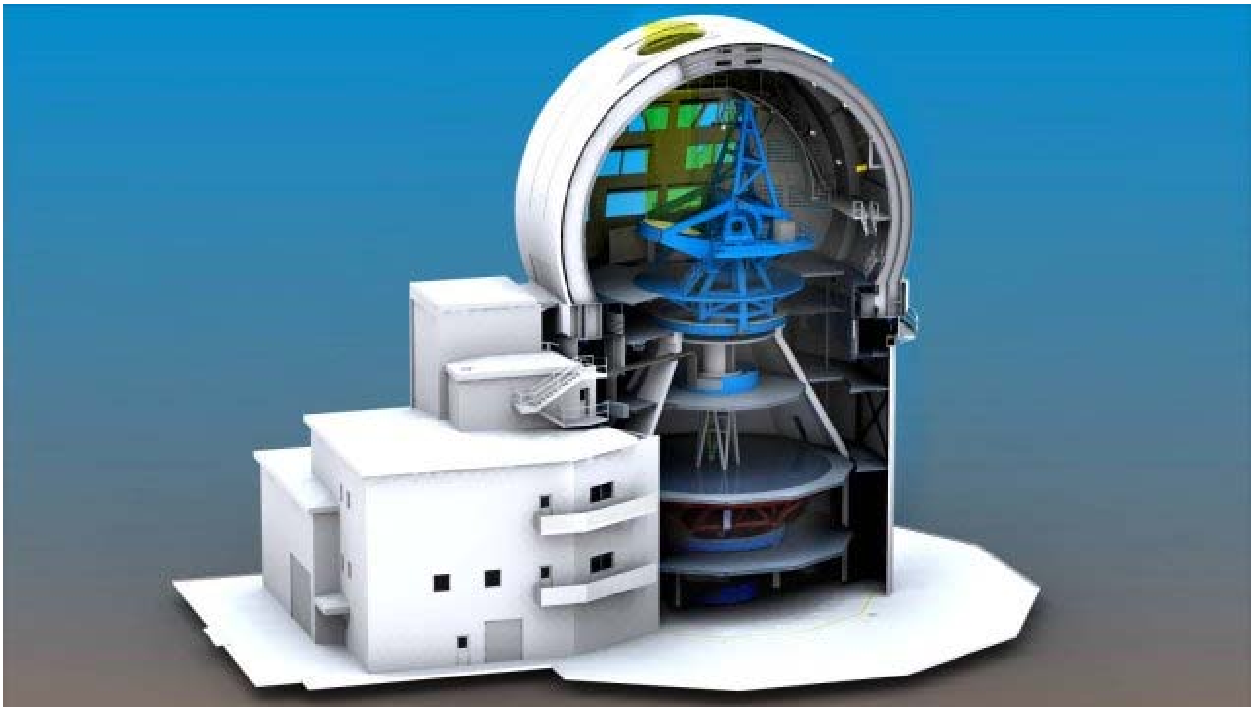

Fig. 1. ATST cutaway view.

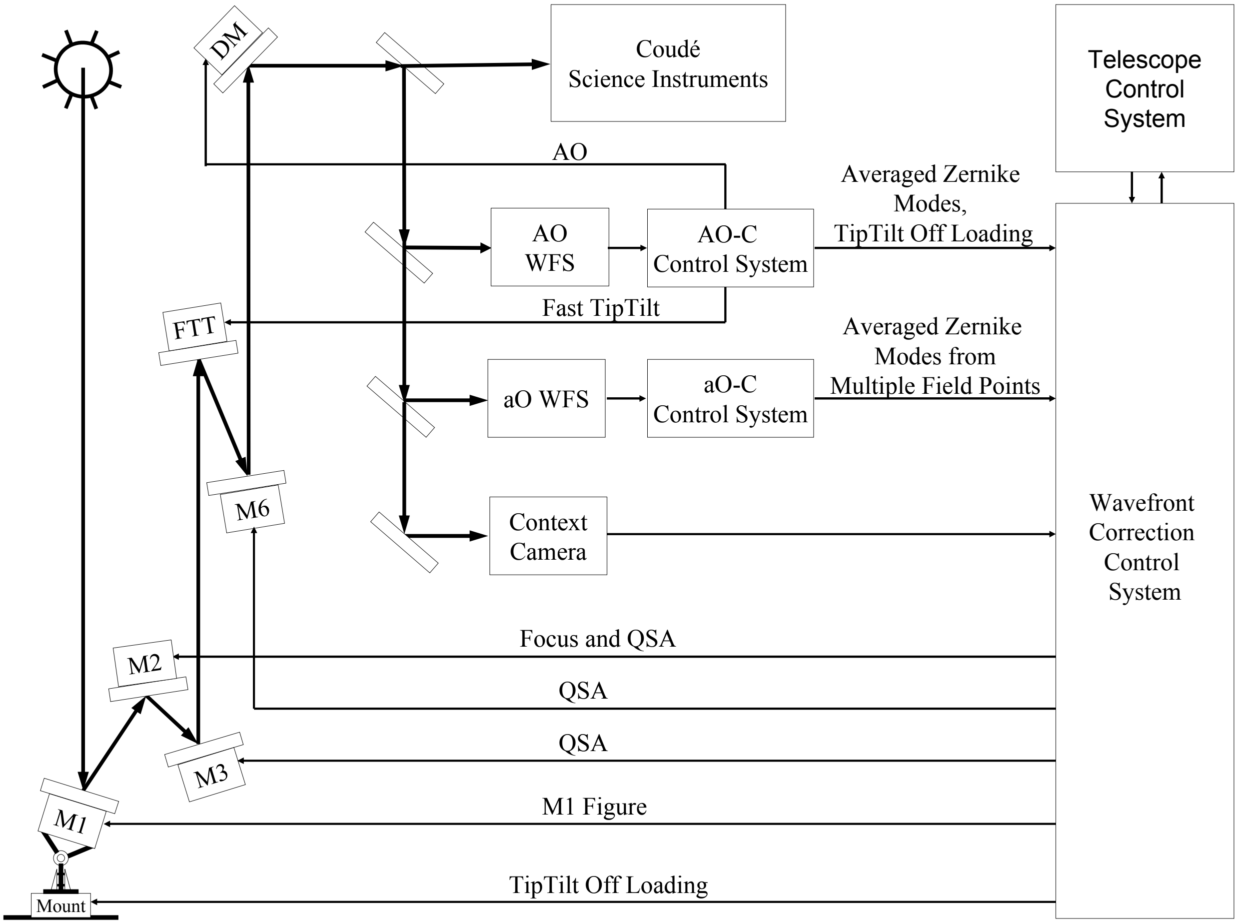

Fig. 2. ATST Wavefront Correction System as interfaced to the light feed of the ATST

The primary function of the AO system is to correct the high spatial and temporal wavefront errors caused by atmospheric effects on the light going to the ATST Coudé science instruments.

The AO system is part of a larger system called the WaveFront Correction System (WFCS) [4]. See Fig. 2.

The WFCS also includes an active optics (aO) system. The aO system's primary function is to sense slowly changing wavefront errors due to telescope flexure caused by the moving gravity vector and temperature changes and to correct these errors by adjusting actuators on the back of M1 and by moving M2, M3 and M6. This function is called Quasi Static Alignment (QSA). [5]

Time averaged aberrations from the AO system and the aO system are combined by the Wavefront Correction Control System which then commands QSA components to correct the aberrations.

2. AO SYSTEM COMPONENTS

The ATST AO system's major components are a lenslet array, a high speed camera, a processing system, a deformable mirror (DM) and a fast tip-tilt mirror (TTM). See Fig. 3. The tip-tilt mirror is used to remove the fast image shift that the deformable mirror doesn't have enough range to do. The deformable mirror changes shape to flatten the incoming wavefront. The Shack-Hartmann lenslet array focuses an array of similar images (subaperture images) of the sun's surface or photosphere onto the sensor of a high speed camera. These images will move in relationship to each other as a result of the earth's atmosphere. The processors read the camera, determine how images are shifted by cross correlations (wavefront sensing), and calculate (reconstruction) the commands for the deformable mirror and tip-tilt mirror to make the needed corrections. This paper focuses on the real-time processing requirements for the ATST AO system.

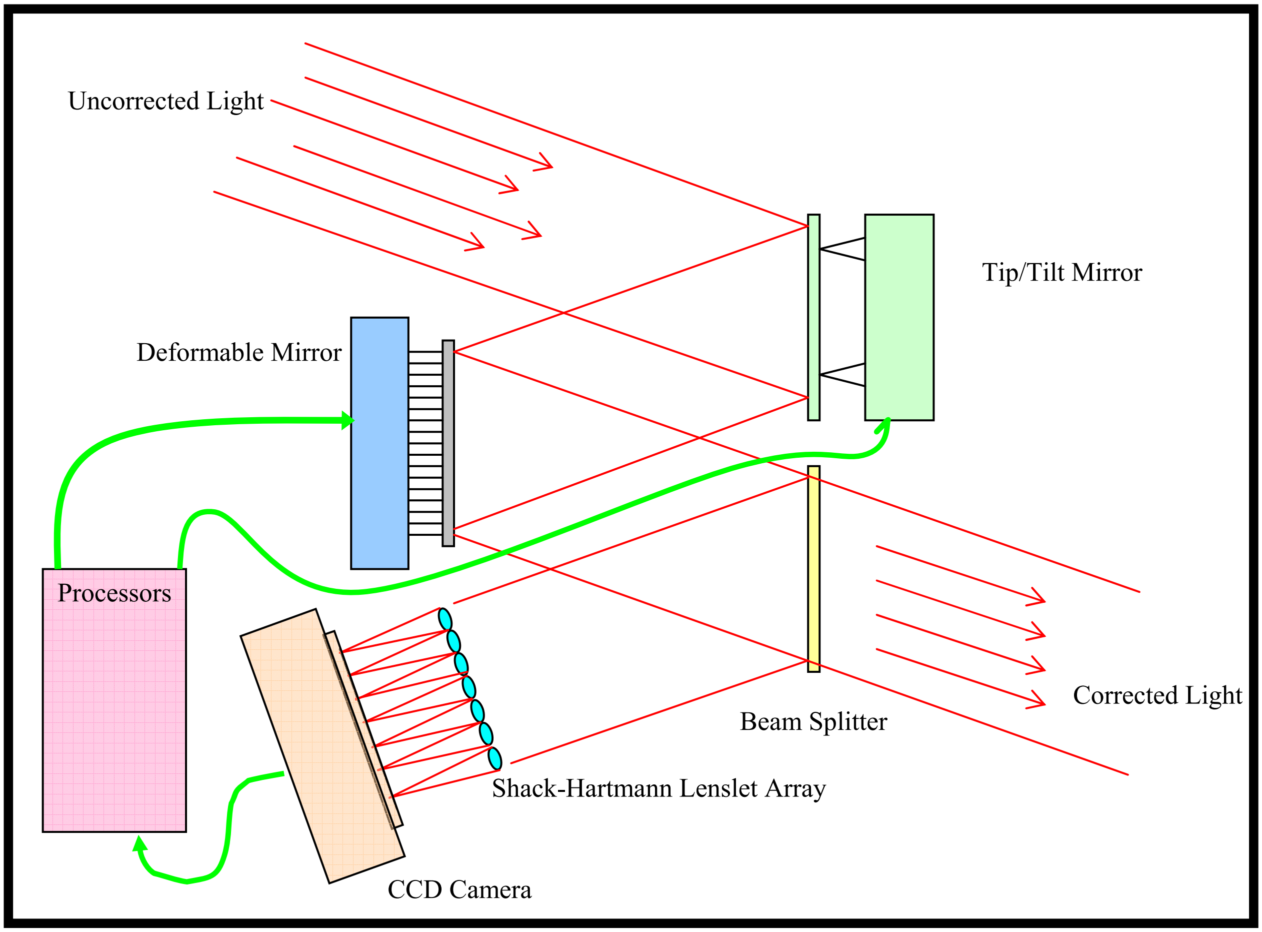

Fig 3. AO system components

3. AO SYSTEM REAL-TIME PROCESSING

Fig. 4 is a block diagram of the real time processing steps for an extended source AO system. The procedure is driven by the frame rate of the high speed camera. As each subaperture image is received, the processing system will apply a dark field correction, a flat field correction and do a 2D cross correlation with the image and a reference image to find the xy shift of the image compared to the reference image. The calibration xy shift values are subtracted and then a partial matrix multiplication is done to calculate the influence of the x and y shift for each of the DM actuators. After all the subaperture images for the frame have been processed, all the partial matrix multiplication products are summed together to give a new value for each of the DM and TTM actuators. A servo loop algorithm is applied to each of these values before they are transmitted to the DM and TTM. This is the same procedure used on the Dunn Solar Telescope AO system developed by NSO and currently regularly used as an user instrument.

The Dunn AO system assumes that the shift between a subaperture image and the reference image will rarely be greater than plus or minus 3 pixels, especially when the system is operating. Therefore only a partial numerical 2D cross correlation is calculated resulting in an array of 7 x 7 values. A sub pixel shift is calculated by fitting a parabolic curve to the maximum value in the 7 x 7 array and the two values on either side in both the x and y direction. The same will be used for the ATST AO system.

For the ATST AO system the following requirements have been determined[4]. There will be approximately 1232 subaperture images in 40x40 rows of 20x20 pixels each. See Fig. 6. This will require a camera with at least 800x800 pixels. The frame rate of the camera needs to be 2000Hz or greater. The DM will have approximately 1313 actuators.

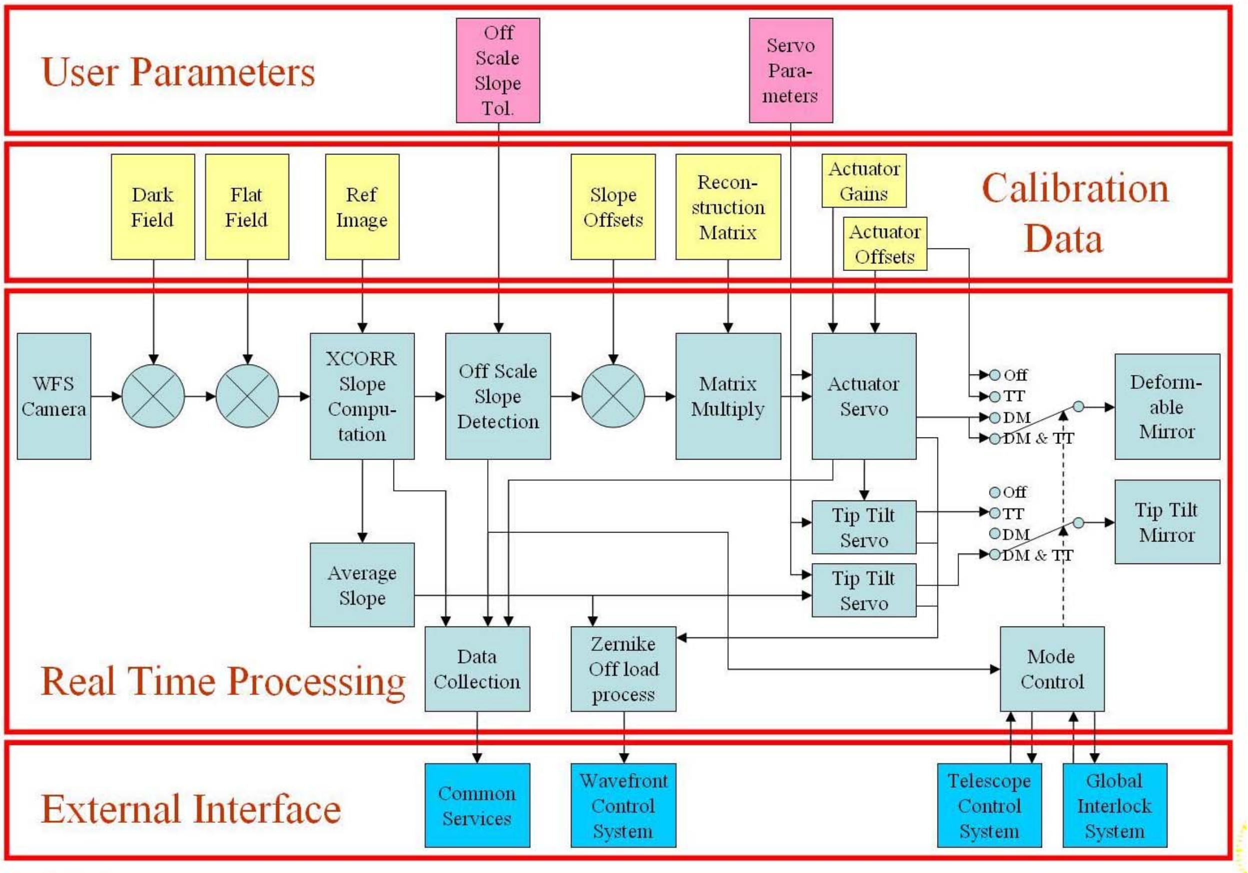

Fig. 4. Block diagram of real-time processing for ATST AO system.

A rough calculation of the minimum processing requirements for each of the 1232 subaperture image is as follows:

Dark field	20 pixels * 20 pixels subtractions	= 400 subtractions
		= 800 memory reads
		= 400 memory writes
Flat Field	20 pixels * 20 pixels multiplies	= 400 multiplies
	20 pixels * 20 pixels shifts	= 400 shifts
		= 800 memory reads
		= 400 memory writes
Cross correlation	20 pixels * 20 pixels * 7 x shifts * 7 y shifts multiply/adds	= 19600 multiply/adds
		= 39200 memory reads
Matrix multiplication	2 shifts * 1313 actuators multiply/adds	= 2626 multiply/adds
		= 2626 memory reads
		= 2626 memory writes

Total operations per subaperture image	= 70,278 operations
Total operations per camera frame = 1232 subapertures * 70,278 operations	= 86,582,496 operations per frame
Total operations per second = 2000 frames * 86,582,496 operations per frame	= 173,164,992,000 operations/sec

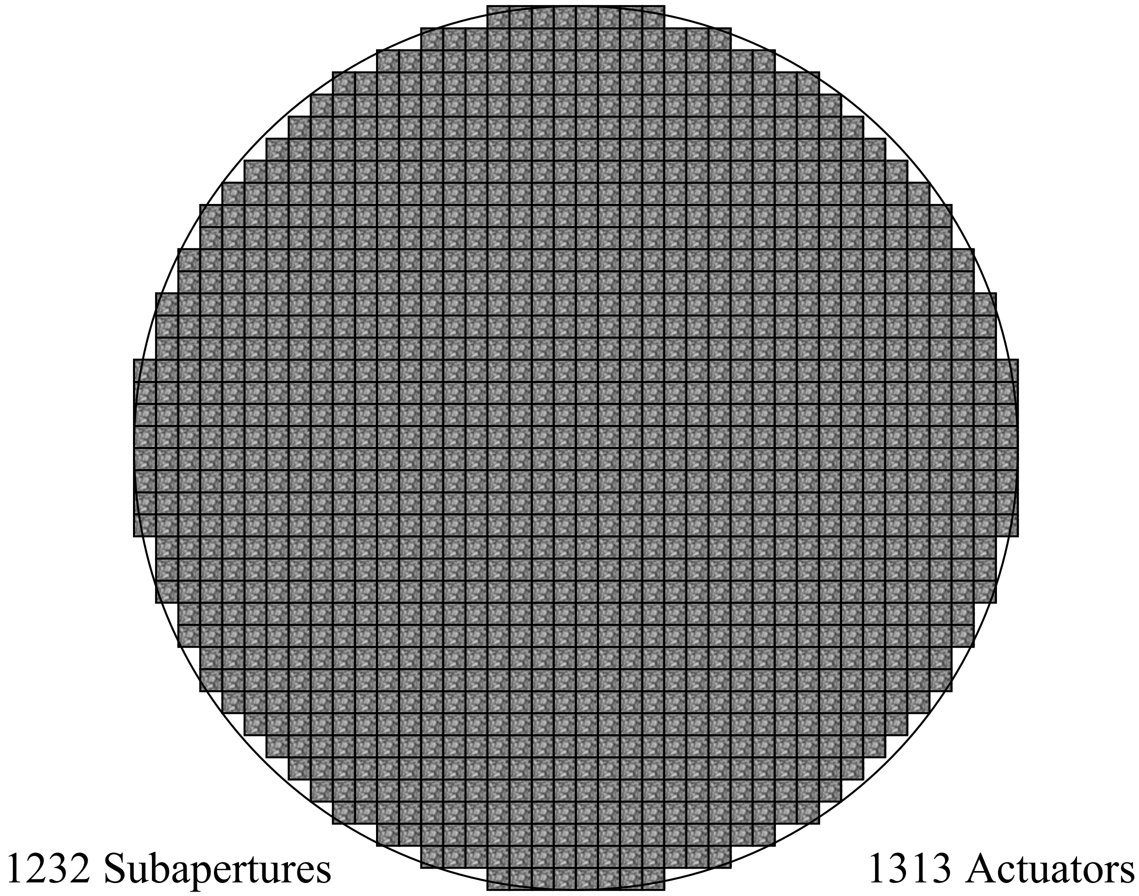

Figure 5. Map of subaperture images on AO camera.

The DST AO system has seventy-six 20x20 pixel subaperture images. The ATST AO system, with 1232 subapertures, will require approximately 15 times the operations per second as the DST. The DST system uses 40 Analog Devices Hammerhead SHARC digital signal processors (DSP) which run at 80MHz and can do 2 math operations in each cycle. The current generation of Analog Devices floating point DSPs is the TigerSHARC [6] which runs at 600MHz and can do eight 16bit integer or 2 floating point multiplies per cycle.

Between the increase in clock rate and greater parallel operations it was estimated that each TigerSHARC DSP could process twenty subaperture images in the same or less time as the Hammerhead DSP could process two. Our current design for the ATST AO system consists of 64 TigerSHARC processors for the real-time processing and up to another 8 TigerSHARC processors to handle the I/O with the camera, DM and TTM.

The DST AO system uses five cPCI boards from Bittware, Inc. with two clusters of four Hammerhead SHARC processors on each board. See Fig. 6. The ATST AO system design includes 8 cPCI Bittware boards [7] each with two clusters of four TigerSHARC processors. Each TigerSHARC cluster also has a Xilinx Vertex II Pro FPGA. See Fig. 8.

The Analog Devices software tools and one cPCI Bittware board were purchased to verify the memory and processing power of the TigerSHARC DSPs would be adequate.

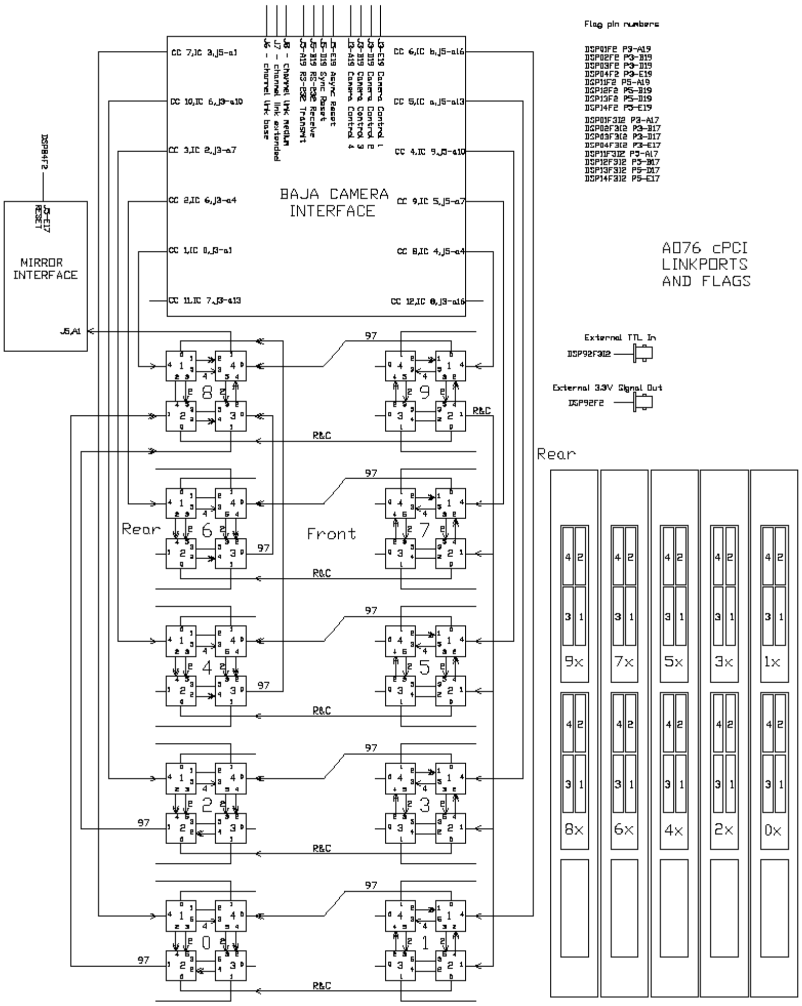

Fig 6. DST AO processing interconnect diagram

4. REAL-TIME SOFTWARE

The AO processing does not require an operating system. The processing is very linear and driven by the frame rate of the camera. There is no need for multiple tasks. There is just one loop and it waits for each subaperture image to be available before it executes. I/O is all done by DMA with interrupts on completion.

The ATST AO real-time software will be written in assembly language, the same as it was for the DST AO. It would have been possible to write the repetitive routines like the matrix multiplication in assembly and the rest in C but doing that would restrict the use of registers and create overhead. Programming everything in assembly language gave complete control of register usage and the ability to write highly optimized code. The Analog Devices development software includes a simulator that makes debugging the assembly code straight forward.

For testing purposes, code for the real-time processing of 20 subapertures in one DSP was first written in IDL running on a PC. IDL has graphical procedures that make it easy to verify that the code is working correctly. The IDL code was written so that it also created data files that can be included by the DSP assembler. The code was written as a series of subroutines – flat and dark field pixel correction, cross correlation and shift calculation, matrix multiplication.

The DSP assembly code was written and tested using the same input data as the IDL routines. The results were compared to verify that the DSP code was generating the same results as the IDL code. A main loop was written with glue code to set up pointers and call each of the subroutines for 20 different subaperture images.

The DSPs have a 64 bit cycle counter that allows precision measurement of the time required to execute a piece of code. This was very useful with the Analog Devices' DSP simulator since the simulator software running on a PC is much slower in time than the same code running on a DSP. The cycle counter also led to the discovery that there are hardware features on the DSPs that decrease the number of cycles needed.

Here is one example. The DSPs' instructions are pipelined. The instruction pipeline is 10 cycles long. The DSPs have a look ahead buffer for conditional branches. A branch can be marked as "most often branching" or "most often not branching". The look ahead buffer sees the branch coming and loads the instruction pipeline with the instructions most likely to be executed. If the other code needs to be executed, there is a 10+ cycle pause while the pipeline is loaded with the other instructions. With the simulator it is easy to see where the processor would need to waste cycles waiting for data and then rewrite the code to make it more efficient.

The TigerSHARC DSPs have six pages of 128K 32bit onboard memory. Each page also has a high speed 4K 32bit cache. There are three 32 bit address/128 bit data busses that each can access any of the memory pages simultaneously. One of the busses is for instructions and the other two for data. Therefore four 32 bit instructions and eight 32 bit data words can be transferred at one time. The assembly language allows specifying up to 4 instructions to be executed simultaneously. There are two sets of computational blocks, each with it's own set of registers. Instructions can be written so that both computational blocks are manipulating data from different memory pages simultaneously. There is also a separate address/data bus for I/O data so I/O can be occurring at the same time as the code is reading and writing other memory pages. See Fig. 7.

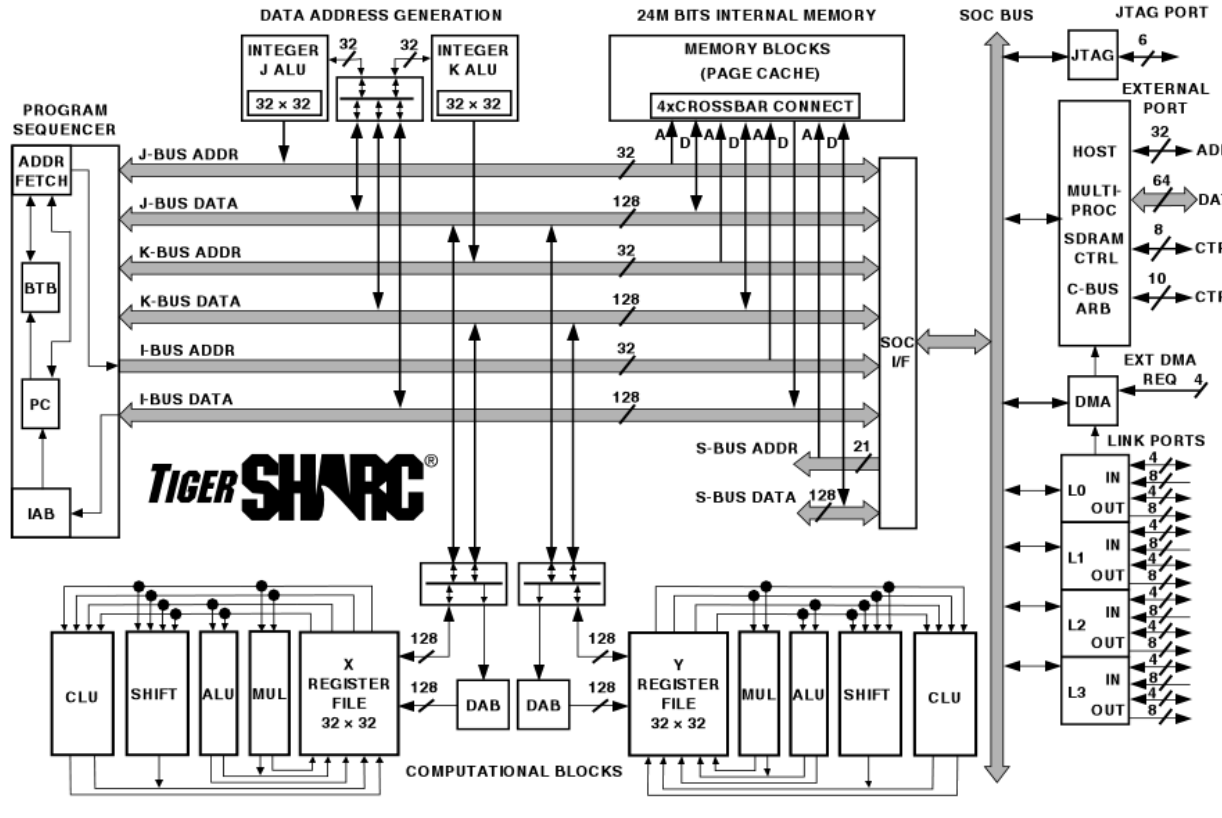

Fig. 7. Analog Devices TigerSHARC TS201 DSP, © Analog Devices, Inc.

The same DSP code was run in the simulator and on real hardware. The two closely agree in number of cycles required. The desired number of pixels for the subaperture images is 20x20. The code for 16x16 pixel images is a bit more efficient to write so both were written and tested. Here is a table of results.

	16x16 pixels		20x20 pixels	
	Cycles	µsecs	Cycles	µsecs
Pixel correction – dark and flat field	310	0.52	469	0.78
7x7 Cross Correlation	3,762	6.27	4,938	8.23
Matrix multiplication	1,461	2.44	1,461	2.44
Total for 20 subapertures (includes some other processing)	112,511	187.52	137,360	228.93

A version of the real-time software was written totally in C and tested with the simulator. The assembly code runs nine times as fast as the C code.

The target frame rate of the camera is 2000 frames/second. This translates to a period of 500 microseconds. The time for a single DSP to process twenty 20x20 pixel subapertures is 230 microseconds, less than half the time available. The remaining time can be used for collecting data such as residual subaperture shift co-variance that can used for estimating point spread functions for image reconstruction. [8]

5. IMAGE DATA TRANSFER

Also of concern is the ability to move image data from the camera to the DSPs at a high enough rate. Assuming 800x800 pixels transferred in 16 bits per pixel at 2000 fps, the data rate is approximately 26 Gbits/second.

For the DST AO system the camera is a custom design with 10 ports. Each port carries eight subaperture images in order. One DSP in a cluster of four receives the eight subapertures images, processes two and passes the other six to the other three DSPs for processing. The data transfers between the DSPs are all done by DMA with no cycles taken from the processor. A similar architecture is intended for the ATST AO system with eighty subaperture images delivered to each cluster of four DSPs each frame of the camera.

Fig. 8 is a block diagram of the proposed ATST AO system. The Smart Interface block design will depend on the camera interface and will most likely be FPGAs. The Smart Interface will sort the pixels from the camera into subaperture image order and feed 20 subaperture images from each frame to each of the DSP clusters.

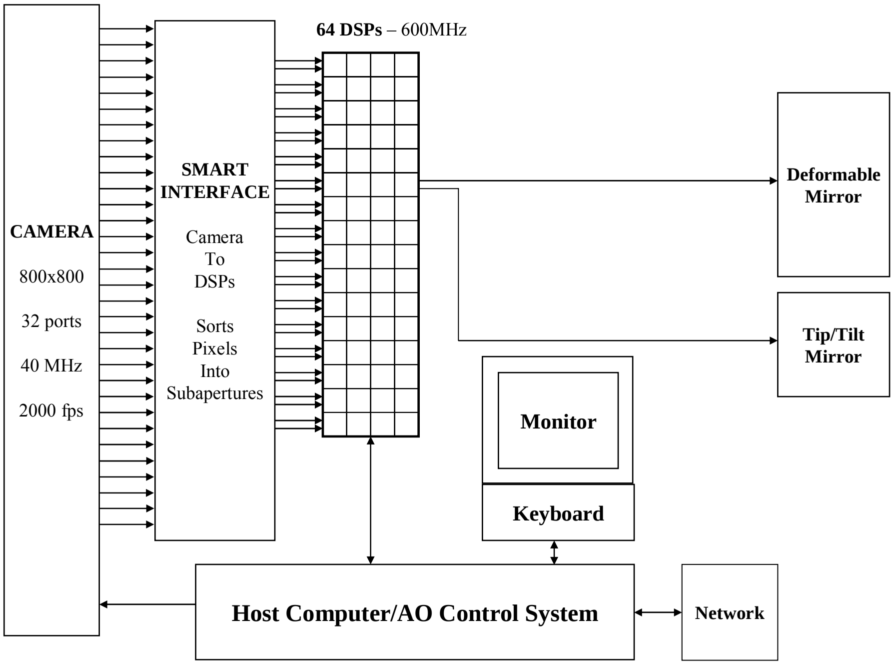

Fig. 8. Proposed ATST AO system block diagram

The Bittware T2-6U boards incorporate two I/O ports per DSP through a Xilinx FPGA Rocket I/O port off board. See Fig. 9. Each of these ports runs 125Mbytes/second. The 20x20 pixel/subaperture image * 2 bytes/pixel * 80 subapertures * 2000 fps equals 128 Mbytes/second - 52% of the available bandwidth of the two ports. This will be verified on the hardware.

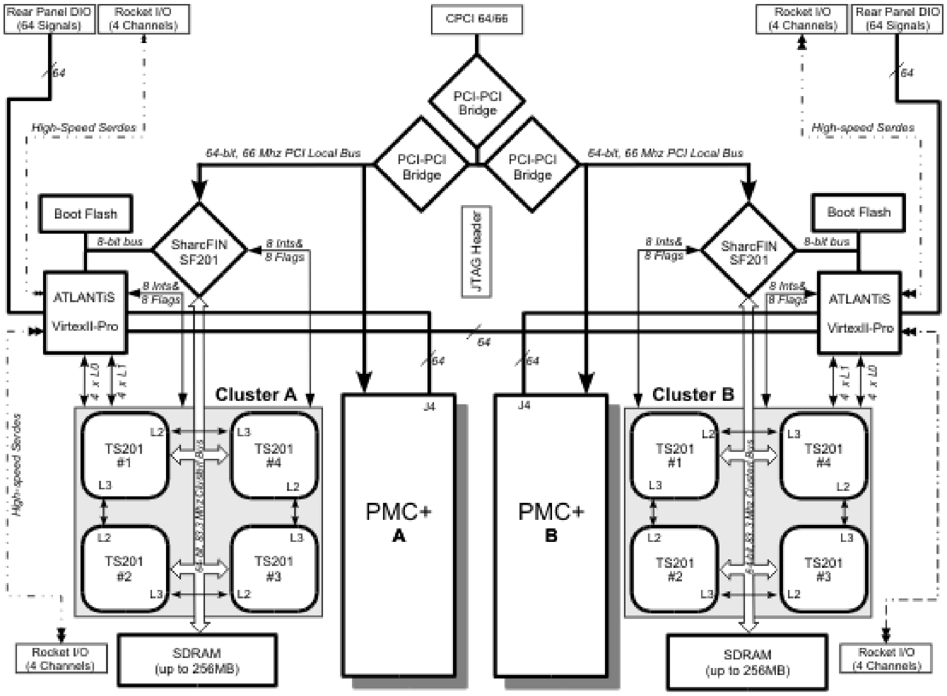

Fig. 9. Bittware, Inc. T2-6U cPCI TigerSHARC board

6. CONCLUSION

Test results on hardware show that real time processing for the ATST AO system can be done by 64 Analog Devices TigerSHARC DSPs in half of the time available. This leaves time for collecting data for things such as point spread function estimation and engineering data for monitoring the performance of the AO system. Further design and testing for camera data transfer will be done.

7. ACKNOWLEDGMENTS

The National Science Foundation (NSF) through the National Solar Observatory (NSO) funds the ATST Project. The NSO is operated under a cooperative agreement between the Association of Universities for Research in Astronomy, Inc. (AURA) and NSF.

8. REFERENCES

1. T. Rimmele, *Recent advances in solar adaptive optics*, SPIE, Vol. 5490,34-46, 2004.
2. S. Keil et al., *Advanced Technology Solar Telescope: conceptual design and status*, SPIE, Vol. 5480, 2004
3. J. Wagner et al., *Advanced Technology Solar Telescope: a progress report*, SPIE, Vol. 7012, 2008
4. T. Rimmele et al., *The wavefront correction system for the Advanced Technology Solar Telescope*, SPIE, Vol. 6272, 2006
5. R. Upton, *Active optical alignment of the Advanced Technology Solar Telescope*, SPIE, Vol 6271, 2006
6. *TigerSHARC Embedded Processor*, Analog Devices, Inc.,Norwood, MA, 2006.
7. Bittware, Inc., *T2-6U-cPCI Octal ADSP-TS201 TigerSHARC board*, Concord, NH
8. Marino, Jose et al., *Long exposure point spread function estimation from adaptive optics loop data: validation and results*, SPIE, Vol 6272, 2006