

## June HASP status report from the SKC Wide Field Camera team

SKC finished its spring quarter classes on June 4<sup>th</sup> and starting June 7<sup>th</sup> eight SKC students have been working on the SKC HASP payload as research interns through NSF funding. Trickel and Olson continue to serve as faculty mentors.

### Accomplishments:

- (1) The direct routing of image data from the image detector to SRAM in hardware has been implemented. As mentioned in the May status report, using C code running on the FPGA to do the pixel-by-pixel write of data to the SRAM is too slow to keep up with the minimum 6 MHz pixel clock of the image detector because the FPGA clock doesn't run fast enough. The hardware logic now implemented on the FPGA can keep up with a pixel clock of up to 12.5 MHz (the SRAM write time is now the limiting factor). We plan to use a 10 MHz oscillator in flight to drive the pixel clock.
- (2) After the new FPGA configuration for direct writing of pixel data to SRAM was implemented we now can acquire uncorrupted images.
- (3) The writing and testing of our flight software is well underway and is on schedule for completion by mid July. The main program flow is a while loop that listens for incoming serial data, either the arrival of a GPS record every minute or an uplinked command. Reception of a GPS record triggers the start of the image acquisition sequence: grabbing an image frame, writing the image data to SRAM, writing the image data from SRAM to the SD card (with a header of a GPS record and temperature data from eight thermistors on the headboard and interface board). The most common uplinked command will be a request for a downlinked thumbnail about every 30 minutes.

### Issues:

- (1) We don't yet have the writing of image data from SRAM to the SD card working. We can get the FPGA to recognize the presence of a SD card in the card socket but the write of data fails. Our design uses the Altera University Program SD card IP core. This IP core can be used in one of two ways, either with direct writes via registers to the SD card without a file system on the SD card, or writes to a file on the SD card within a FAT16 file system through a hardware abstraction layer. For the direct write approach a 512-byte buffer register is filled with image data, the address of the 512-byte block on the SD card to be written to is written into the command argument register, and the write is triggered by writing the WRITE BLOCK command to the command register. We have the register writes working but the writes to the SD card fail, and the status registers for the Altera AD card IP core flag some errors. We are using these flagged errors to find the source of the failed writes.
- (2) Without having the SD card writes working yet we currently view our image data by writing the data from SRAM to a file on a lab computer over a serial connection. The image data captured to the file usually has a few (one to a half dozen) extra bytes of data more than it should for the image size. This seems to mostly be from serial transmission errors (we're transmitting this data without error correction). But some images have an extra byte that was added to the SRAM during image acquisition. These extra bytes are easy to identify and remove to get an uncorrupted image, but they are worrisome because it indicates something about our system we don't fully understand.

### Overall status:

We have a few important issues yet to resolve before integration in August but we are on schedule.