Paper ID: 53099
student

26th IAA SYMPOSIUM ON SMALL SATELLITE MISSIONS (B4)
Small Earth Observation Missions (4)

Author: Mr. Parth Kalgaonkar
India

Mr. Neelanchal Joshi
India

DESIGN OF PAYLOAD DATAFLOW IN A HYPERSPECTRAL IMAGING NANOSATELLITE

Abstract

Team Anant, a group of undergraduate students from BITS-Pilani, is building a nanosatellite whose objective is to perform hyperspectral imaging of the Indian Ocean. The spectral distribution will be used to categorize various phytoplankton and their quantity in the ocean. Because of the low bandwidth availability for data downlink on nanosatellites and the large image size, the image must be compressed by the On-Board Computer. Image capture and compression are both power-intensive tasks. Due to the power constraints in a nanosatellite, such power-intensive tasks must be scheduled at appropriate times when there is enough power. This mandates that the image must first be stored onboard a memory device. Push broom scanners additionally require the image to be stitched before any compression can start. Our On-Board Computer(OBC) is based on the Zynq-7000 SoC, which internally contains two ARM Cortex A9 processors, a Field Programmable Gate Array (FPGA) fabric and various peripheral controllers. The ARM Cortex A9 processor is running a Linux Based Operating system on it which executes the software to control other functions of the satellite. One of the tasks it executes is the stitching of the image before compression. The image compression algorithm is implemented on the FPGA to improve its performance. In this design, we were faced with two major problems. First, the Payload and Image Compression algorithm access the data on the flash memory using physical addresses, while the Zynq PS running an OS accesses it using virtual addresses. In the interaction between the payload and the OBC, the design must come up with an appropriate method strategy to convert the physical addresses to virtual addresses. Then, during the interaction between the OBC and the FPGA, the virtual addresses must be converted back to the physical addresses. The second problem that we faced was the memory interface data transfer rates which proved to be a major speed bottleneck between the memory and the FPGA. This paper explores the various design strategies to counter these two problems. It also compares different types of memory interface controllers such as SPI, CAN, UART, etc implemented on the FPGA to access the image from the memory in terms of throughput, power requirements and ease of implementation in hardware. This paper also discusses various techniques for image stitching and how to implement it on the Zynq-7000 SoC as part of the design.