

The Use of Field Programmable Gate Arrays (FPGA) in Small Satellite Communication Systems

Kosta Varnavas, William Herbert Sims, Joseph Casas

National Aeronautics and Space Administration , NASA / MSFC / ES33, Huntsville, USA

Kosta.varnavas@nasa.gov , Herb.sims@nasa.gov, joseph.casas@nasa.gov

Abstract— This paper will describe the use of digital Field Programmable Gate Arrays (FPGA) to contribute to advancing the state-of-the-art in software defined radio (SDR) transponder design for the emerging SmallSat and CubeSat industry and to provide advances for NASA as described in the TAO5 Communication and Navigation Roadmap. The use of software defined radios (SDR) has been around for a long time. A typical implementation of the SDR is to use a processor and write software to implement all the functions of filtering, carrier recovery, error correction, framing etc. Even with modern high speed and low power digital signal processors, high speed memories , and efficient coding , the compute intensive nature of digital filters, error correcting and other algorithms is too much for modern processors to get efficient use of the available bandwidth to the ground. By using FPGAs, these compute intensive tasks can be done in parallel, pipelined fashion and more efficiently use every clock cycle to significantly increase throughput while maintaining low power. These methods will implement digital radios with significant data rates in the X and Ka bands. Using these state-of-the-art technologies, unprecedented uplink and downlink capabilities can be achieved in a ½ U sized telemetry system. Additionally, modern FPGAs have embedded processing systems, such as ARM cores, integrated inside the FPGA allowing mundane tasks such as parameter commanding to occur easily and flexibly. Potential partners include other NASA centers, industry and the DOD. These assets are associated with small satellite demonstration flights, LEO and deep space applications. MSFC currently has an SDR transponder test-bed using Hardware-in-the-Loop techniques to evaluate and improve SDR technologies.

Keywords - *Software Defined Radio; Field Programmable Gate Arrays; Programmable Lightweight System Adaptable Radio, PULSAR; Finite Impulse Response Filter; microprocessor; digital signal processor; parallel processing.*

I. INTRODUCTION

Marshall Space Flight Center (MSFC) has been developing a low-cost software defined radio transponder which contributes to advancing the state-of-the-art in telemetry system design which is directly applicable to the Small Sat and CubeSat communities. The SDR, called PULSAR – Programmable Ultra Lightweight System Adaptable Radio, can be incorporated into orbital and suborbital platforms.

By examining a number of the systems available for current CubeSats, they do not have sufficient bandwidth or processing capability for transmitters and receivers to

support new error correcting protocols as well as innovative payload designs with complex encryption schemes being developed by the CubeSat community (academic, military, civil, industry). The PULSAR SDR has a highly efficient



Figure 1. Ground Station Antennas

SWaP, (Size, Weight and Power), which achieves higher bits per input supply watt (at ~10 Mbits per input watt) than traditional communication SDR systems (at ~300Kbits per input watt) requirements.

The paper will go into the details of a technical approach to implement high speed flexible satellite radios. It will then compare the differences between implementing these techniques using processors or FPGAs, and then it will look at how this work aligns with the NASA roadmap.

II. TECHNICAL APPROACH

A basic SDR block diagram is shown in figure 2 [1]. As is typical, the concept of the SDR is to minimize the analog / radio frequency (RF) components and do as much as possible in the digital domain.

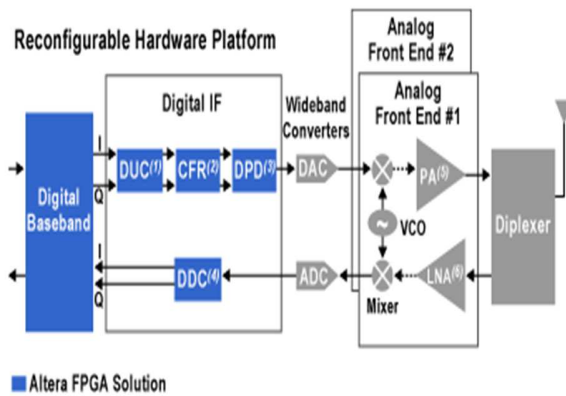


Figure 2. Typical Software Defined Radio [1]

Many traditional software defined radios actually use processors to accomplish these tasks. With low bandwidth requirements and frequency bands below the L-Band range, 1-GHz processors can handle the workload. Even with minor forward error correction coding, the processors can still handle the load. However, as the frequencies climb into and above S-band, 2.0 GHz, and data rates increase significantly, even fast digital signal processors will have trouble keeping up with executing all the code necessary to do filtering, digital up converting and down converting, as well as forward error correction schemes such as Reed-Solomon, Low Density Parity Check and others. Add encryption of any type to the mix, and the processor will get bogged down quickly. Utilizing multiple processors or even multi-core processors are all advanced means of achieving the throughput necessary. However, the complexity of these systems grows as will the cost, size and power.

Because of the above problems with utilizing digital signal processors, Marshall Space Flight Center’s SDR, PULSAR, chose to use Field Programmable Gate Arrays such as the Actel ProAsic3 Flash devices. All signal processing algorithms are done inside the FPGA and designed using Hardware Description Language (HDL).

The PULSAR radio is divided up into a series of stackable decks. This can be seen in Figure 3. Each one stacks on top of the other to make a very modular system that can be customized for each mission’s requirements. Each deck is designed to be stand-alone with the exception of the power deck, for any configuration. However, even the power deck can be eliminated if filtered, isolated power of the right voltages are provided. Of the stackable decks available, the S-band transmitter will transmit Quadrature Phase Shift Keyed (QPSK) data at 5-10 Mbps. The X-band transmitter will transmit one channel of QPSK data at 110 Mbps, and the S-band receiver will receive data at 300 kbps. Although, maximum data rates on the uplink have not been tested, it is believed that at least 1 Mbps could be uploaded in the current hardware and FPGA algorithmic configurations.

These data rates are not the limit of the hardware or of the algorithms inside the FPGA. They are the constraints placed on the satellite transceivers by the NASA Near Earth

Network (NEN)[5]. In addition to these radio component decks, there is a power deck to provide isolated power to the entire stack. Also, there is a processor deck that utilizes an embedded ARM processor inside of the FPGA. This can be used for additional algorithms or as the flight computer itself. With an embedded ARM processor [6]and external ram, the processor deck has enough computational power to be a flight controller for many applications.

Because each deck can be stand alone, each deck has its own FPGA and performs all signal processing inside of the FPGA. This provides plenty of extensibility to allow additional functions and algorithms to be added to each deck’s. The FPGA chosen for this version of the PULSAR is the Actel ProAsic3 [7]. It is a flash based FPGA which means it is live at power up and does not need a configuration memory like SRAM based FPGAs such as Xilinx.

Exemplifying the flexibility of PULSAR, transmission can occur using Low Density Parity Check (LDPC), Reed-Solomon (255/223), or convolutional (Rate ½) Forward Error Correction (FEC) codes based on mission requirements. Each of these codes, except the Rate ½ convolutional encoding, is very compute intensive. The intensity of computations necessary to implement these FECs limit a digital signal processors ability to perform these functions and maintain a high data rate to the ground.

A unique feature of the power deck is its ability to monitor current draw on each of the power rails going to each of the decks. The power deck is programmed for a maximum current draw per deck and when it is exceeded, the power rail is turned off until the fault is cleared. This is designed in as a radiation mitigation method to prevent radiation induced latch-up.

Each of the radio decks, transmitters and receiver, have a number of digital algorithms it has to perform. The S-band receiver has algorithms it has to perform to recover the signal and strip the data of headers and error correction to get to the actual message / commands sent. The transmitter decks have algorithms to perform Forward Error Correction (FEC), and NEN compatible packetization. This is typical of SDRs and is what provides their small size and low power and flexibility. However, the amount of processing involved becomes more difficult to do in a processor the higher the data rates and frequency. Filters, carrier recovery loops, error correcting decoding are all very compute intensive.

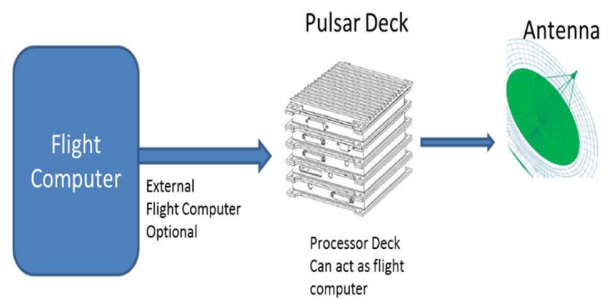


Figure 3. PULSAR stack in system. Flight computer optional.

III. PERFORMANCE

Computations take so much more time in a processor versus an FPGA. Processors of any kind, even those designed specifically for signal processing, can only perform one instruction per clock cycle. And that is for the fast and well-designed processors. But even so, some computations require more than one, maybe many clock cycles to achieve one action. For a receiver, if a processor is reading the analog to digital (A/D) converter it may take several instructions to point to the A/D, read the data and store it in a register. Cache and direct memory access will help, but it gets worse. Now that you have the data, you may want to down convert it again which uses a numerically controlled oscillator and digital mixers which are multiplier heavy. Then it will need to be filtered. Filters have numerous multiplies and adds. All this has to be done in a seamless, continuous manner to get the data to come through correctly. A Finite Impulse Response Filter (FIR) is a typical type of filter for signal processing of all types. A typical digital signal processor (DSP) such as the TI TMS320C55x at 300 MHz built for signal processing, has direct memory access, cache and embedded hardware multipliers. These features enhance data throughput. Memory is treated as a circular buffer with pointers that automatically update. Due to a high amount of parallelism and pipelining, the whole - fetch data, perform MAC, return data - process is usually done in one clock cycle.

Loop X times:

- Move (2) input samples from memory to MAC
- Move coefficient from memory to MAC
- Perform MAC operation

Retrieve output from MAC and store in memory
Send output sample to interface.

The higher order the filter, the higher number of taps are necessary and the number of loops, X, goes up significantly.

Using the features of the above processor, a 422 tap FIR filter could be implemented up to ~ 628 kbps. But this will come at a cost of ~ 200mW of power! Additionally, processors and clock rates are not linearly correlated. Just because one DSP had a higher clock rate, it doesn't mean it will perform better on a benchmark.

So with the above example, there is a lot of overhead for circuit design, and power to achieve less than 1 Mbps on just one FIR filter. In a typical SDR there will be numerous filters as well as other digital algorithms and multiple data paths in the case of mPSK modulations schemes. So why do we not just run to faster DSPs? There is a limit to the speed and the power hit goes up linearly! See Figure 4 [2].

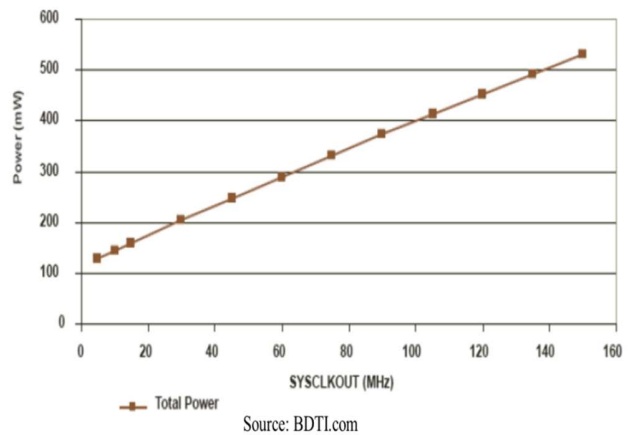


Figure 4. Power Consumption vs. Clock Frequency [2]

In an FPGA, all of these functions are performed in a pipeline manner. In addition, each step of the pipeline has all the adders, multipliers necessary to accomplish the task. The data is presented to the next stage on every clock cycle, eliminating the need to fetch data and instructions as to what to do with the data, on every iteration. There are still numerous functions that require numerous iterations, but with the pipelined structure and some parallelism, the main clock does not have to be as high to achieve the same data rates. Additionally, some of the newer FPGAs have numerous dedicated hardware multipliers. This speeds up the multiplication process itself, which can be an iterative implementation.

Figure 5 [3] shows a graphic explaining the problem using a general-purpose digital signal processor.

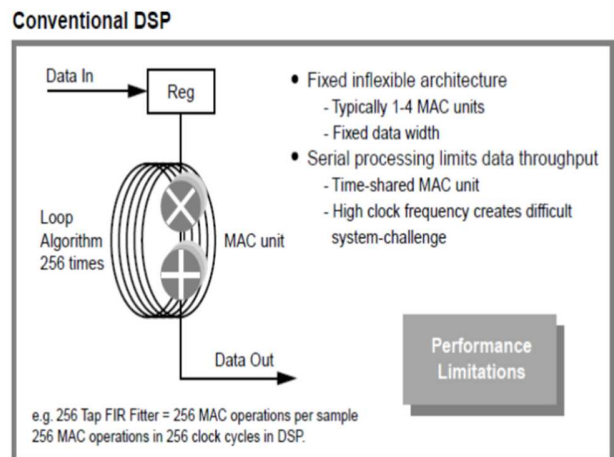


Figure 5. Conventional DSP Implementation [3]

Even the latest processors still have the above limitations. However, as Figure 6 [3] shows, an FPGA clearly has the ability to perform, parallel, pipelined functions with local dedicated hardware, from multipliers to block rams, enabling a much more efficient use of the clock and at a much lower overall clock speed.

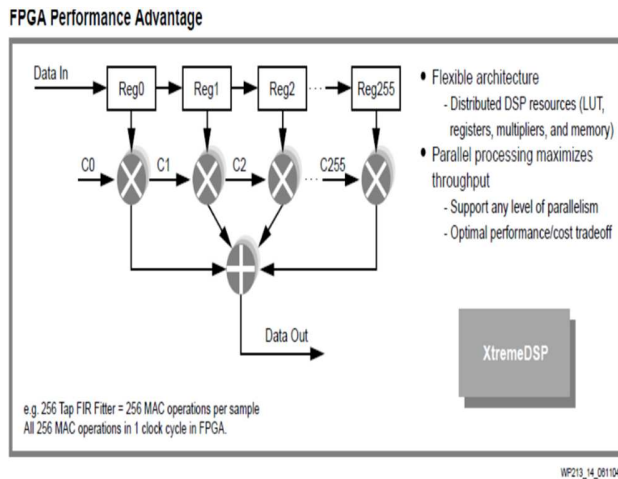


Figure 6. FPGA Performance advantage. [3]

IV. ALIGNMENT

NASA is called, at the direction of the President and Congress, to maintain an enterprise of technology that aligns with missions and contributes to the Nation's innovative economy. NASA has been and should be at the forefront of scientific and technological innovation. In response to these calls, NASA generated a plan (NASA Strategic Space Technology Investment Plan [4] to advance technologies and nurture new innovation that will feed into future missions. PULSAR aligns primarily with the Technology Area (TA) 5 – Communication & Navigation – but has connections to other areas in which lightweight structures, power efficiency, and communication reliability and throughput are the focus.

V. CONCLUSION AND FUTURE WORK

Currently PULSAR implements an S-band transmitter, S-band receiver and X-band transmitter utilizing advanced FPGA technology and digital signal processing techniques inside the FPGA. As a complete integrated unit, PULSAR has been tested in a lab environment with typical NEN ground station equipment, procedures and operational

scenarios. Upcoming builds of this system are planned for full environmental testing. This includes Electromagnetic Interference/compatibility tests, Thermal/ Vacuum tests, as well as vibration tests. A variety of potential upcoming flights will allow PULSAR to fly as a payload to prove its capability as a flight read instrument. An upcoming ground demonstration with the Space Launch System could be its first relevant environment testing. This test will interface numerous development flight instrumentation (DFI) sensors with the PULSAR to transmit to the remote test station. This will eliminate significant amounts of long run cabling.

Depending on funding levels, future developments of the next generation of software defined radios on the roadmap, include a C-band transceiver, an X-band receiver to complement the current X-band transmitter, and at some point Ka-band transponders. The PULSAR team is constantly applying the latest innovations to provide cutting edge systems for small satellite communications systems.

REFERENCES

- [1] http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCQOFjAB&url=http%3A%2F%2Fweb.ccs.pdx.edu%2F~mperkows%2FCAPSTONES%2FDSP1%2Fpresentation%2520of%2520Software%2520Defined%2520Radio.ppt&ei=GNG2VKzdAcbksATJgYLoBA&usq=AFQjCNHRWPr0lcC8Njnr5rQocs5HSiB1A&sig2=XQw0-5wjEJ_qg3HbHQY7fAL-3 Communication Systems-West, "Cadet Nanosat Radio," Product specification sheet, pp. 6, 2011, retrieved: January, 2015
- [2] QuickfilterTechnologies, Inc., "Digital Filtering Alternatives for Embedded Designs" <http://www.quickfiltertech.com/files/Digital%20Filtering%20Alternatives%20for%20Embedded%20Designs.pdf> pp. 16, September 2006, retrieved: January, 2015
- [3] Xilinx, http://www.xilinx.com/support/documentation/white_papers/wp213.pdf, Xilinx White Paper, pp. 18, July 2004, retrieved: January, 2015
- [4] National Aeronautical and Space Administration, "NASA Strategic Space Technology Investment Plan" NASA Washington, DC, 2013. pp. 1 – 92, December 2012, retrieved: January, 2015
- [5] Nasa Near Earth Network, https://www.nasa.gov/directorates/heo/scan/services/networks/txt_nen.html
- [6] ARM Processors, <http://www.arm.com/products/processors/cortex-m/index.php>
- [7] ActelProAsic3, <http://www.microsemi.com/products/fpga-soc/fpga/proasic3-overview>