Pater ID: 272697

Compressive Sampling and Real-Time Data Transportation in MPAR Backend System

Xining Yu[1], Rockee (Yan) Zhang[1], Mark Weber[2,] and Allen Zahari[2]

[1]Intelligent Aerospace Radar Team, Advanced Radar Research Center/School of ECE, University of Oklahoma

[2]NOAA OAR National Severe Storms Laboratory, and University of Oklahoma Cooperative Institute for Mesoscale Meteorological Studies

Abstract: With the extensive usage of advanced waveforms, the bandwidth of the transmitting pulses can be quite large. To handle those large amount of data rate, a flexible and efficient fabric interconnects technology, RapidIO, has been used as the backplane communication protocol. In order to further reduce the data stream rate without loss of information, we propose to incorporate compressive sampling (CS) concept into the MPAR backend system.

A small-scale DSP-based computing testbed is established to verify the CS channel compression concept and emulate end-to-end computing latency. Firstly, Matlab simulations has been performed for both discrete targets and distributed targets. The simulated data is downloaded to the computing platform, and real-time pulse compression, Doppler processing as well as beamforming are implemented. The channel performance and overall latency with and without CS is compared.

I. MPAR Backend System Modeling

Three different types of backend boards are used in the fabric, and as the data moves more towards the back, more DSPs are utilized. The boards are: (1) Receiver front-end/data transmitting board, (2) DSP processing board, and (3) Back Panel. With current low-cost COTS components, the data transmitting board can capture up to 40 independent channels, and these signals are digitalized by ADC and packed up by FPGAs. On the back panel, 12 RapidIO switches can handle up to 2.8 Tera-bit per second data rate. The materials costs for an actual implementation of a 200-channel backend will be less than $400K.

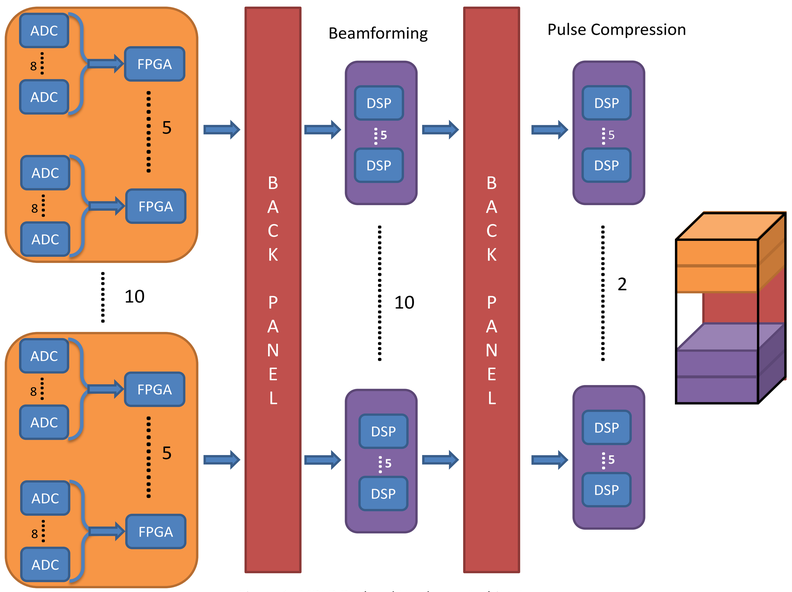

Figure 1: MPAR Backend Hardware Architecture

II. Small Scale HPEC for Real-Time Basic Processing Demonstration

In a small scale model, we simulated realistic sampling data traffic from RF front end to Beamforming DSP, as well as Beamforming DSP to Pulse Compression DSP. In this system, we have 24 C66x DSP cores, and 4 ARM cores to handle 24 channels with 8192 range gates. The data coming from 24 channels will be generated into 20 beams by 16 DSP cores, and then 8 DSP cores is responsible for Pulse Compression and Doppler Processing. ARM cores would handle the data traffic between Pulse Compression and Doppler Processing. Assuming Nyquist sampling speed at front-end, the estimated transmission time from a front end FPGA to Beamforming DSP is 904 µsec for one pulse.

One of the key challenges to achieve hard-real-time using embedded DSP processors is how to assign tasks to each DSP core. As the hardware resources on DSP core is limited, Beamforming weight vector is calculated from outside and transmitted to DSP core. For the rest of beamforming, which multiplies each channel's samples by weight vectors, DSPs needs ~1.5 msec. After that, Pulse Compression needs another 1.5 msec to complete. The data transmission between Beamforming and Pulse Compression is done by using EDMA (Enhanced DMA), which could send or receive the data without interfering with DSP. With help of EDMA, calculation can be performed in parallel with data communication. As 20 beams cannot evenly divided into 8 DSP cores, so some of DSP cores would be free from Pulse Compression task. Doppler Processing is performed on those stall cores during this period. The data corner turn, in which rearrange range-aligned data to pulse-aligned, is performed by ARM core using EDMA, which costs 1.1 msec. The following Figure 2 sketches the real-time time-line for initial radar processing. Further processing will be then added, such as weather product generation and target tracking.

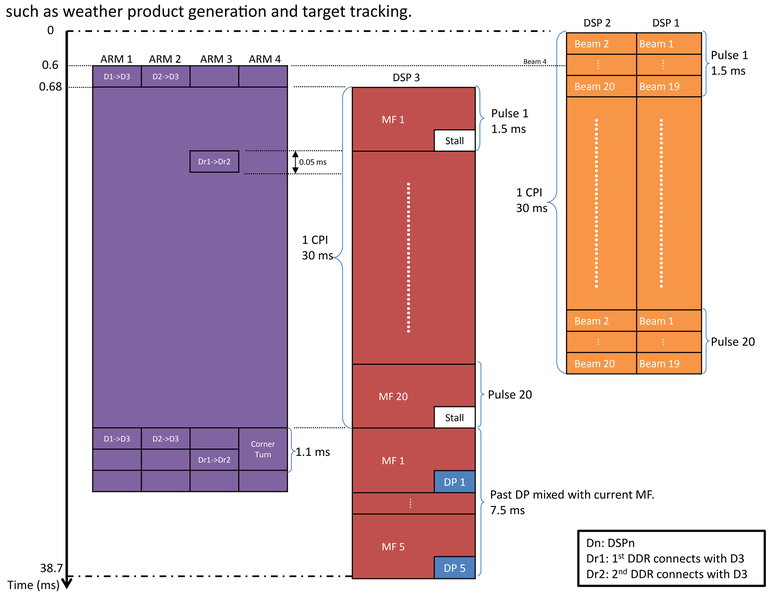

Figure 2: Small scale system timeline

According to the timeline, the shortest PRI (Pulse Repetition Interval) being supported is ~ 1.5 msec, or 667 KHz for PRT. Faster processing can be achieved by adding more parallel computing hardware (similar to the boards shown as in below). Calculation load of Doppler Processing is much less computational demanding compared to the Beamforming and Pulse Compression.

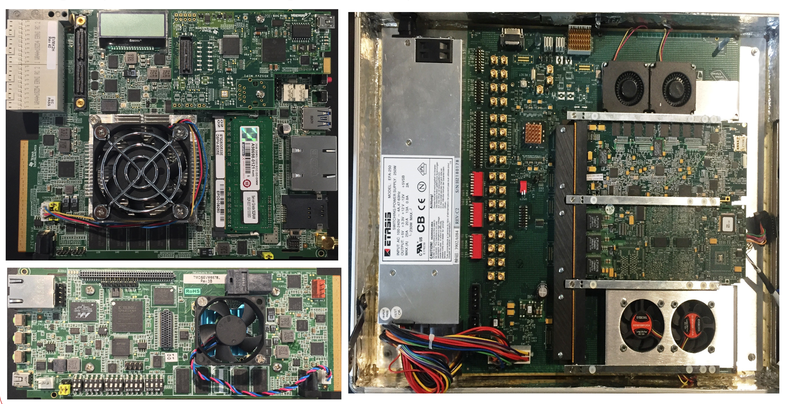

Figure 3: Processing Board

III. Compressive Sampling

In order to further reduce the data stream rate without loss of information, we propose to incorporate compressive sampling (CS) concept into the MPAR backend system. This novel sampling method challenges the commonly well know Nyquist sampling rate that requires the sampling speed higher than two times of signal bandwidth, and it can recover certain signals by using fewer samples. For the concise representations of original signal, two fundamental premises should be met: sparsity and incoherence. For the sparsity, it means that when the signal is projected onto a suitable basis, a large number of coefficients of signal should be small enough to be ignored. For a certain signal, if it has s non-zero coefficients, it is said to be s-sparse. As s increases, it becomes harder to sense and reconstruct the original signal. The incoherence implies that any two elements in the sensing basis Φ and representation basis Ψ should have low coherence. The coherence between Φ and Ψ is measured by

$$\mu(\Phi,\Psi) = \sqrt{n} \cdot \max_{1 \le k,j \le n} |\langle \phi_k \psi_j \rangle|$$

The sampling and representation basis should be concerned as low coherence pairs. For example, we may choose spike basis as sensing matrix, and Fourier basis as representation basis.

Array channel simulation parameters

Parameter	Values
Pulse bandwidth	5 MHz
Pulse width	10 µsec
PRF	6 KHz
Nyquist sampling rate	10 MHz
CS sampling rate	5 MHz
Number of range gates	1700
Number of range gates for LFM pulse	102

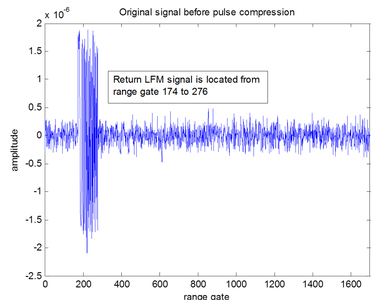

Figure 4: Simulated point target return (LFM waveform, I samples only) before pulse compression, SNR= 9 dB.

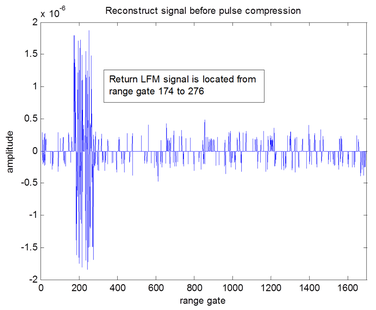

Figure 5: Reconstructed point target return from signals with ½ original sampling rate (5 MHz).

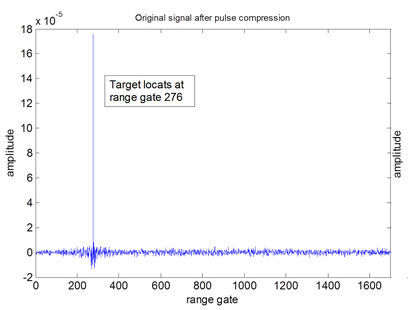

Figure 6: Simulated point target return (amplitude) after pulse compression, pre-compression SNR= 9 dB.

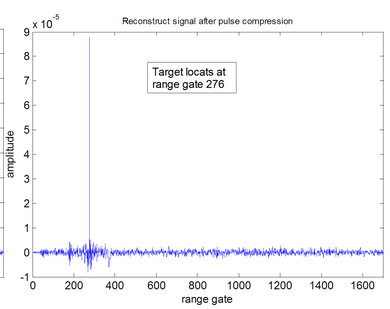

Figure 7: Reconstructed pulse compression output from signals with ½ original sampling rate (5 MHz).

Contact: rockee@ou.edu Tel: (405)3256036

This work is supported by NOAA-NSSL through grant #NA11OAR4320072. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Ocean and Atmospheric Administration.

School of Electrical & Computer Engineering