A RECONFIGURABLE 100 MCHIP/S SPREAD SPECTRUM RECEIVER

Jon Arnold, Adrian Caldow, Kevin Harman

Defence Science and Technology Organisation
Cooperative Research Centre for Satellite Systems
PO Box 1500, Edinburgh, South Australia, 5111
E-mail: jon.arnold@dsto.defence.gov.au

ABSTRACT

The flexibility of a reconfigurable receiver allows a system to be tailored to maximise the data rate over channels that vary from benign to those severely effected by multipath propagation. Accordingly a digital receiver based on Field Programmable Gate Arrays (FPGA) has been developed, with a burst-mode 100 Mchip per second (100 Mcps) Direct Sequence Spread Spectrum (DSSS) personality that targets a 10Mbit/s data rate in extreme multipath scenarios. The design is based around a quadrature, two sample per chip baseband processing card that implements an asynchronous feedforward (interpolating) architecture. With this approach, system complexity resides solely in the receiver allowing the transmitter to be very small and cheap. This paper briefly reviews the architecture and implementation of the receiver, and reports on some of the development and performance issues.

1. INTRODUCTION

The system under research and development is an S-band wireless digital communications link utilising either differential BPSK (DBPSK) or differential QPSK (DQPSK) DSSS modulation with spreading rates up to 100 Mcps. Operation is targeted at the deployment of simple, commercial grade, portable transmitters for either terrestrial or satellite based communications that are received and demodulated using a remotely controlled and reconfigurable receiver via a standard network connection. Initially the link will operate in a point-to-point environment either terrestrially or via a bent pipe geosynchronous X band or Ka Band satellite transponder.

The primary goals of the project are to implement wideband acquisition, tracking, despreading and demodulation schemes, and achieve improvements in receiver flexibility and performance through high levels of application specific digital signal processing (DSP) within FPGAs. The wideband DSSS processing will facilitate either high data rates of the order of 1 to 10 Mbit/s in a harsh fading (multipath, doppler) channel, with burst transmissions and variable data rates, or lower data rates of the order of 10 kbit/s for even poorer environments or where improved communications security or antijam capability is required. Remote control and monitoring is achieved by designing the receiver circuit cards as slaves on a cPCI-bus hosted by a single board computer running the Linux operating system. This, and the reconfigurability of the digital baseband processing, provide the functionality to not only remotely configure, monitor and test a given implementation, but also to reprogram the receiver architecture for different operational scenarios.

Section 2 of this paper will outline the motivation and selection of the overall receiver architecture. Section 3 then highlights aspects of the firmware DSP. Finally, Section 4 presents performance issues.

2. RECEIVER OVERVIEW

The receiver architecture can be coarsely partitioned into an analogue or RF ‘front end’ and a digital baseband processor. These have been implemented on two separate PCI-controlled printed circuit boards, one containing the RF sections of the receiver and the remainder the DSP functions. This paper focuses on the DSP board.

A traditional approach for processing the DSSS waveform would be based on a matched filter acquisition circuit for rapid PN code alignment and some form of Digital Delay Lock Loop (DDLL) for chip tracking, where the timing adjustment is applied using synchronous feedback that controls the ADC sample clock. With our high maximum chip rate of 100 Mcps the design of both the Voltage Controlled Oscillator (VCO) and the loop itself becomes very difficult. This is further exacerbated by the requirement for a burst communications protocol, which demands rapid lock and settling times. Another complicating factor is the digital delay that would be incurred within the synchronous DDLL, which from previous experience could be tens of chips and make loop stability difficult to achieve [1].

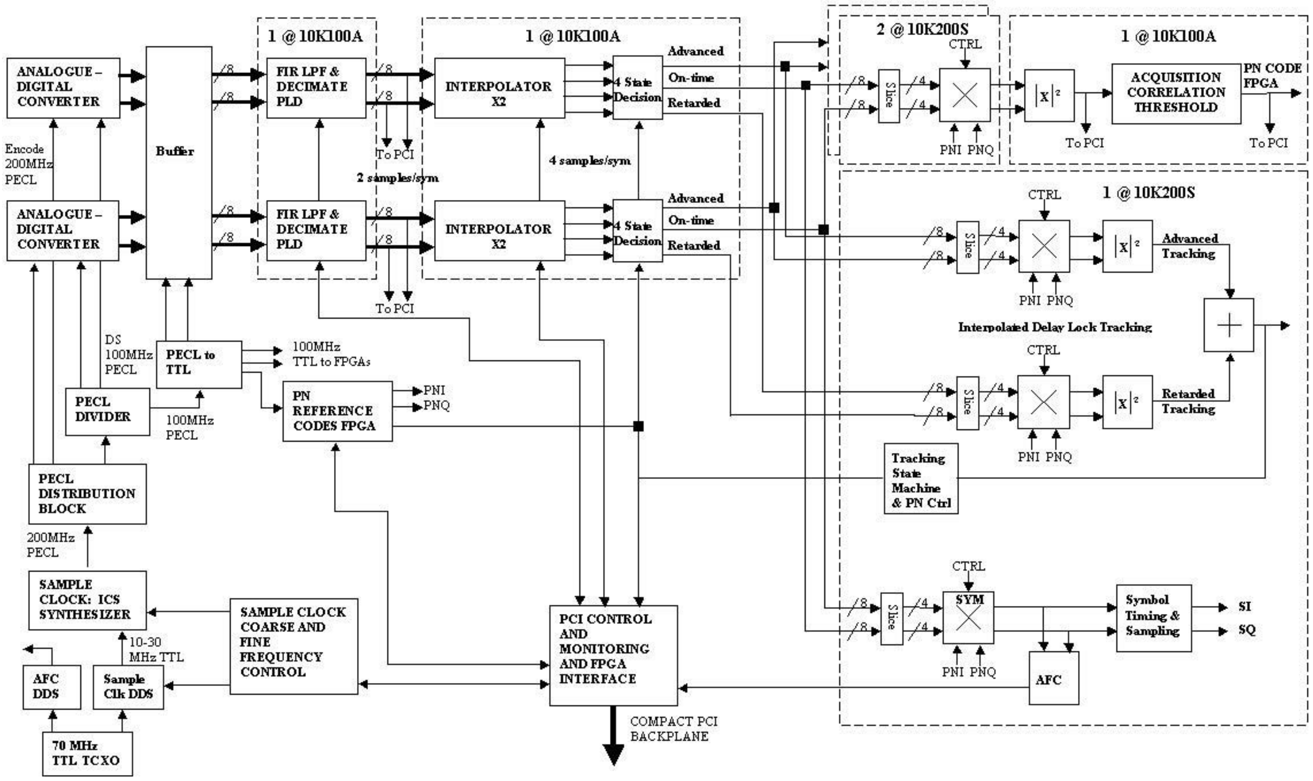


Figure 1 Receiver Baseband Architecture.

To avoid this an asynchronous feed forward (A-FF) architecture has been chosen. This removes the need for a VCO and allows for a fully digital implementation without the long feedback delay.

Essential to the success of the A-FF approach is the need for sufficient oversampling of the input signal, thereby permitting at least one of the samples to be near to the optimum (chip-centred) point without any feedback timing adjustments. Sufficient oversampling (x4 or x8 (400 or 800 Msps)) could be achieved with high rate ADCs, but this necessitates ECL-technology devices with associated size and power penalties in buffering the ECL levels into the (CMOS) FPGAs.

The preferred alternative was to use slower rate CMOS ADCs and improve the effective sampling rate by interpolation. This leads to the architecture that is depicted in Figure 1, in which two 200 Msps, 10-bit ADCs provide quadrature sampling at the minimum (x2) oversampling rate and digital interpolators are used to increase the effective sample rate to x4. Each ADC output is in a parallel multiplexed form of two streams at 100Msps, which is compatible with the maximum processing rate of the chosen Altera FLEX10K100A and FLEX10K200S FPGAs.

3. BASEBAND FPGA DESIGNS

As apparent from Figure 1, the primary DSP performed in the FPGAs consists of filtering, interpolation, DSSS acquisition and tracking, DSSS despreading, PSK demodulation and control. These are detailed further below. First however, it is noteworthy that the 100 Mcps chip rate, selected to attempt to realise high processing gains, posed a considerable implementation challenge. Fitting of the synchronous, register-intensive algorithms in the FPGAs required significant optimisation for both resource conservation and to achieve the desired throughput.

Chip filtering after the ADC is provided by a root Nyquist FIR filter. The design is unconventional in that a multi-rate structure has been used to allow the 200 Msps input data to be processed at the 100 Msps FPGA rate. Odd and even input samples are processed by individual 100 Msps FIR filters with relative input and output timing adjustments applied to give the equivalent 200 Msps filtered output. Additionally, each filter has a multiplierless implementation to conserve logic resources. Overall, two 17 Tap, 18 Bit filters have been produced for each quadrature branch. The filtered samples then enter the interpolator.

The *interpolators* produce on-time, advanced and retarded samples for accurately tracking timing offsets of the received pseudo-noise (PN) spreading codes.

Simulations comparing linear and cubic interpolator structures were undertaken and showed that only a small improvement could be realised by implementing a five state cubic interpolator compared to a four state linear version. As a result a 4 state (1/4 chip tracking resolution) linear interpolator was implemented. This buffers 5 half-chip resolution samples and then interpolates between pairs of adjacent samples to obtain a set with quarter chip resolution. The required samples from the set are selected under the control of the DDLL.

For *acquisition* the design assumed that the receiver would be operating at a negative chip SNR, and hence would require acquisition before frequency control [2]. This necessitates a *non-coherent detector* where the I and Q channels must be jointly detected. Initially a suboptimum technique was applied, where the correlation sum is evaluated as:

$$[(R_i*PN_i) + (R_q*PN_q)]^2 + [(R_i*PN_q) - (R_q*PN_i)]^2 \quad [3]$$

where R_i and R_q are the respective I and Q chip samples and PN_i and PN_q the receivers replica PN codes. Future work will involve the development of a detector whose performance is not degraded by the underlying data modulation.

The FPGA implementation of the detector takes the form of a multi-mode parallel correlator, which may be applied in a parallel or serial manner. The parallel modes are QPSK x 1-bit (512 chips long), 2-bit (256 chips) or 4-bit (128 chips) and BPSK x 1-bit (1024 chips) or 4-bit (256 chips). The serial mode allows an accumulation dwell of up to 512 times in the selected format. A detector exists for both the on-time and advanced samples, each implemented in a separate FPGA, with utilisations of 97% and throughput exceeding 100MHz.

Detector outputs are processed by an *acquisition controller* in the form of an (up to) 8 level acquisition dwell tree, with independent thresholding at each dwell test [4, 5]. The detector correlators are initially loaded with known PN_i and PN_q phases and held static. Threshold testing then begins as R_i and R_q are clocked past these references, and the dwell tree state is incremented for each successful threshold test. On the first successful test acquisition is declared and the replica PN codes no longer remain static but are stepped in phase with the input samples. Acquisition is held during transitions up and down the dwell tree until the bottom is again reached where the static phases are reloaded and the acquisition process repeated.

Once acquisition is declared the *tracking loop* is enabled. This performs non-coherent complex correlations against both the advanced and retarded input streams and the difference between the two correlations is smoothed and scaled to produce points on the classic early late gate tracking ‘S-curve’. A controller then uses this error magnitude to adjust the interpolator state and select the best available sample to within ¼ chip resolution. The tracking loop correlators are implemented using the serial ‘accumulate and dump’ structure, and have programmable dwell lengths. Algorithmically, these correlators are similar to those used in the acquisition loop, however they have been simplified to $[(R_i*PN_i)^2 + (R_q*PN_i)^2]$ to compensate for the nulling problem due to symbol modulation. In a future revision, the detector will be modified to exploit the total chip energy.

Symbol timing recovery has been implemented for the generic case of an arbitrary number of chips per symbol, not the easier case of one PN code iteration per symbol. A real, non-coherent correlation against the I-channel on-time received samples is performed using a serial correlator started at the assumed symbol mid-point and reset after the correct symbol dwell (in chips per symbol). For the case where the symbol polarity changes during the correlation (accumulation), the correlator output will tend to zero if the assumed timing phase was correct, otherwise the output magnitude is proportional to the timing error. To correctly process the direction of error, and inhibit loop updates when no polarity change occurred, a second correlation synchronised to the assumed true symbol timing is also performed and permits estimation of symbol polarity.

The incoming samples are then *despread* for symbol recovery using the on-time I and Q samples, the local PN_i and PN_q and the known symbol timing from above. The complex symbol Y is estimated noncoherently by:

$$Y = Y_i + j*Y_q \text{ where}$$

$$Y_i = (R_i*PN_i) + (R_i*PN_q) \text{ and}$$

$$Y_q = (R_q*PN_q) + (R_q*PN_i).$$

The local PN codes are correlated with the four most significant bits of I and Q on-time and an output symbol in rectangular form is produced. A rectangular to polar conversion is then performed to simplify the *differential detection* process. This is done using a multiplierless 6-stage CORDIC rotator [6], with 8-bit angle resolution. The differentially detected symbols Y_d permit a hard-decision and output of the transmitted serial data.

The final significant subsystem is the automatic frequency control (AFC) loop. Phase error estimates are produced for every symbol by multiplying the differentially decoded symbol phase by four and averaging this four times phase error over a block of symbols using a resource-saving exponential averaging approach. The effective averaging window is approximately 30 symbols. Negligible losses are incurred from using the first order symbol Y_d instead of the optimal detector based on Y_d^4 [7].

Figure 2 is a photograph of the completed receiver, showing (from left to right) a test modulator card, the baseband FPGA processing card, and the S band downconverter card. Six FPGAs providing approximately one million gates of reconfigurable logic form the heart of the baseband processing card. Figure 3 shows outputs of the receiver's graphical user interface (GUI). Configuration, monitoring, data acquisition and post processing such as power spectral estimation, can all be controlled through the GUI.

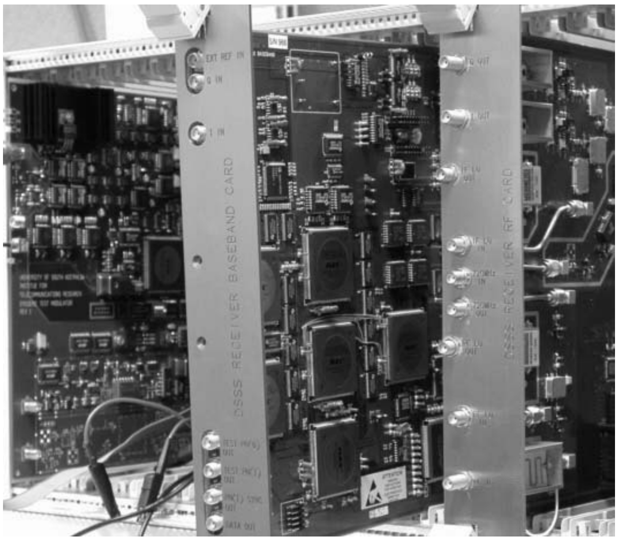
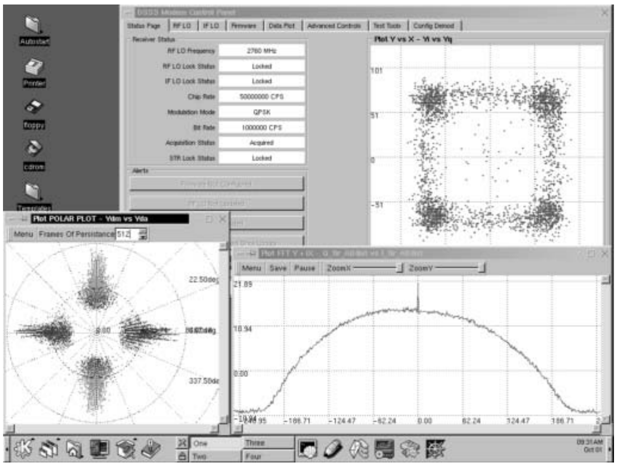


Figure 2 Test Modulator, Baseband and RF cards.



RF LO Frequency: 2760 MHz
RF LO Lock Status: Locked
IF LO Lock Status: Locked
Chip Rate: 50000000 CPS
Modulation Mode: QPSK
Bit Rate: 1000000 CPS
Acquisition Status: Acquired
STR Lock Status: Locked

Frames Of Persistance: 512

Figure 3 Graphical User Interface.

4. PERFORMANCE & CONCLUSIONS

Initial characterisation testing of the receiver has been conducted at 50 Mcps, with underlying data rates of 50 kbps to 2 Mbps, and for chip SNR as low as -20dB. Results for individual circuits, including acquisition detector performance versus chip SNR and frequency error, tracking loop S-curve gain and tracking loop bandwidth versus gain, show excellent correspondence with theoretical and simulated assessments. Results for the end-to-end demodulation performance, as measured by the BER with respect to the ideal DQPSK BER curve, show a consistent 6dB implementation loss, which is attributable to several components. Briefly, these losses include a 3dB loss due to noncoherent combining in the despreader circuit and a 1 to 2dB loss due to the A-FF tracking and interpolation process. A +/-1dB uncertainty in the 0dB SNR reference point may also contribute.

Test and development work is ongoing, and future reports will provide more detailed descriptions of test results, including wireless testing in true multipath channels, and of the identification and optimisation of implementation losses.

5. ACKNOWLEDGEMENTS

The Authors wish to thank Professor Bill Cowley and Dr. Jinho Choi of the Institute for Telecommunications Research, University of South Australia for their contributions in helping to define and analyse the initial system architecture [8].

This work was carried out with financial support from the Commonwealth Government through the Cooperative Research Centres Program.

6. REFERENCES

[1] B.Cowley, B.Moran, M. Bottema, *Analysis of Phase Locked Loops with Large Delay*, Engineering Mathematics and Applications Conference, 1998.

[2] R. L. Peterson, R. E. Ziemer, D.E. Borth, *Introduction to Spread Spectrum Communication,* Prentice Hall, New Jersey, 1995.

[3] A.J. Viterbi, CDMA, Principles of Spread Spectrum Communication, Addsion Wesley, New York, 1995.

[4] A. Polydoros, C. L. Weber, "A Unified Approach to Serial Search Spread Code Acquisition-Part I & II", IEEE Trans. Comm., Vol. 32, No. 5, May 1984, pp 542-560.

[5] E. A. Sourour, S. C. Gupta, "Direct Sequence Spread-Spectrum Parallel Acquisition in a Fading Mobile Channel", IEEE Trans. Comm., Vol. 38, No. 7, July 1990, pp 992-998.

[6] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers", *Proc.* of *the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays,* Feb. 22-24, 1998, Monterey, CA. pp 191-200.

[7] A. J. Viterbi, A. M. Viterbi, "Nonlinear Estimation of PSK-Modulated Carrier Phase with Application to Burst Digital Transmission", IEEE Trans. Information Theory, Vol. IT-29, No. 4, July 1983, pp 543-551.

[8] W.G. Cowley, J. Choi "B1.5 DSSS Modem Project Stage I Final Report", Institute for Telecommunications Research, Adelaide South Australia, 1999.