DVB-S2 Software Defined Radio Modem on the RC64 Many-core Digital Signal Processing

S Pradeep Kumar
Asst Professor,Department of ECE,Kakatiya Institute of Technology and Science,Warangal,TS,India

Abstract—

This paper describes high performance implementation of DVB-S2 modem on the rad-hard manycore RC64 DSP. Multi-level simulation and development methodologies are described. Modem algorithms are specified, together with implementation details. Efficient parallel processing is enabled by the shared memory architecture, by PRAM-like task oriented programming and by dynamic allocation of tasks to cores. The modem achieves in excess of 2 Gbps transmission and 1 Gbps reception.

INTRODUCTION

RC64 is designed as a high performance rad-hard manycore DSP processor for space applications [1][8]. The architecture is shown in Figure 1. 64 DSP cores (CEVA X1643) are integrated together with hardware accelerators, a hardware scheduler, multi-bank shared memory, a logarithmic network on chip connecting the cores to the memories.

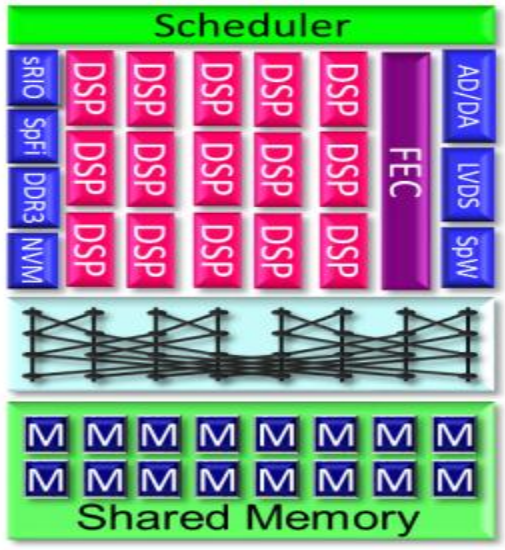

Figure 1. RC64 Many-Core Architecture. 64 DSP cores, modem accelerators and multiple DMA controllers of I/O interfaces access the multibank shared memory through a logarithmic network. The hardware scheduler dispatches fine grain tasks to cores, accelerators and I/O.

The paper presents the simulator, the modem algorithms, implementation details, parallel programming of the model, and performance evaluation

Software Defined Radio (SDR) and modems constitute very demanding applications. This paper investigates the implementation of DVB-S2/DVB-S2x modems on RC64. An LDPC hardware accelerator is included in RC64 to support efficient modems, and as a result RC64 achieves in excess of 2 Gbps transmit rate and 1 Gbps receive rate. Earlier works in this area include [6] and [7]. The RC64 DVB-S2 modem has been developed using a multi-level methodology and simulators. The development of a modem on a manycore processor combines communication theory, parallel algorithm design, parallel programming and profiling, and software engineering

RC64 DVB-S2 SIMULATOR

Figure 2 depicts the RC64 DVB-S2 simulator structure. The data generator creates baseband frames. The transmitter encodes and modulates the frames according to DVB-S2 and DVB-S2X standards. The channel simulator adds noise and impairments. The receiver demodulates

and decodes the signal, and the analyzer compares the sent and received signals.

The simulator enables testing and performance optimization regarding modem quality (bit error rate for a range of channel impairments, signal to noise ratio and bandwidth), modem bitrate (performance of RC64 executing the modem application), bottleneck analysis (identify required accelerator(s) for the modem) and hardware accelerators type and capacity (validation before hardware integration).

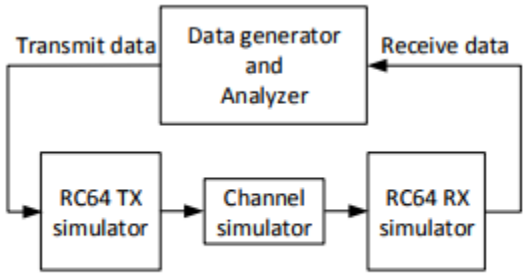

Figure 2. RC64 DVB-S2 Simulator

Modem development is carried out through six levels of refinement, as shown in Table 1. Algorithm development starts by coding in Matlab a high level model of the modem, and proceeds through stages until finally parallel C code is employed to program the actual RC64. We start with an unrestricted algorithm, implemented in Matlab (level 1). The accelerators code is replaced by a Matlab executable (mex) file generated from RTL descriptions of the accelerators. Level 1 serves as golden model, to which subsequent level models may be compared. Level 2 takes into account architectural restrictions of RC64 such as limited memory and real-time constraints. For instance, receiver input samples are processed in pre-defined sample groups rather than in frame size sample groups. In the third level, Matlab floating-point computations are replaced by Matlab fixed point at a word precision of 16 bits, compatible with high-speed arithmetic on the DSP cores of RC64. Accelerator models are replaced by more precise ones driven from RTL. Outputs are carefully compared with the results of the floating-point models, to assure minimal signal degradation. At level 4, Matlab is replaced by code in the C language, compatible with the compiler for the DSP cores in RC64. The Matlab simulator models of the transmitter and receiver are replaced by models for the cycle accurate simulator of RC64. The output must be exactly the same as produced in level 3. The accelerator code is a function in C representing the hardware accelerator. At level 5, the code is parallelized to execute on RC64 and further optimizations are performed to take advantage of specific hardware features of the DSP cores. The accelerators function is executed as a separate task, in parallel with other tasks. In level 6 the entire modem is executed on RC64 hardware

Table 1. Levels of Simulation and Modem Development

Level	Level Name	Language	Precision	Style	Accelerators
1	High Level Modem	Matlab	Float	Virtual unlimited architecture	FloatC-to-mex
2	Matlab DSP Modem	Matlab	Float	Restricted to real-time DSP of RC64 Restricted memory sizes Translate input frames to samples on TX, input sample stream to frames on RX.	FloatC-to-mex
3	Fixed Point Matlab DSP Modem	Matlab	Fixed 16	Rounding and saturated computation Use CEVA lib functions	RTL-to-mex
4	C-Fixed Modem	C	Fixed 16	Bit-exact to Level 3	C function
5	C-Parallel Modem	C	Fixed 16	Compliant to Plural shared-memory programming model [8]	C function as a separate task
6	RC64 Modem	C	Fixed 16		Task on accelerator hardware

RC64 DVB-S2 MODEM ALGORITHMS

In this section we describe the algorithms of the transmitter, the communication channel, the receiver and the data generator and analyzer. Transmitter

The DVB-S2 and DVB-S2X transmitter includes the following functional blocks to modulate input streams, as specified and recommended in [2][3][4] (Figure 3): CRC-8 encoder, baseband (BB) header insertion and stream adaptation, BB Scrambling, FEC encoding (comprising BCH and LDPC encoders and bit interleaver), bit mapping into constellations, physical layer framing (PL header insertion, followed by pilot adding and scrambling) and BB shaping

(up-sampling and low-pass filtering). This series of functional blocks can be clustered into Pre-LDPC stage, the LDPC encoder, and Post-LDPC stage

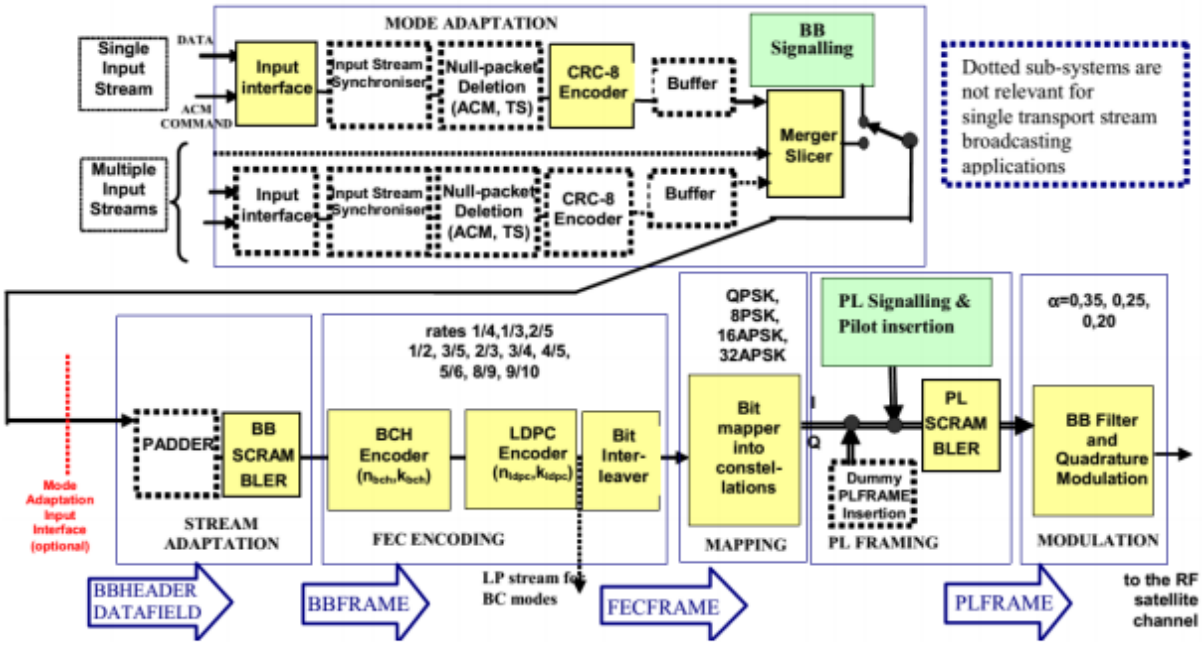

Figure 3. Functional block diagram of the DVB-S2 transmitter (following [3])

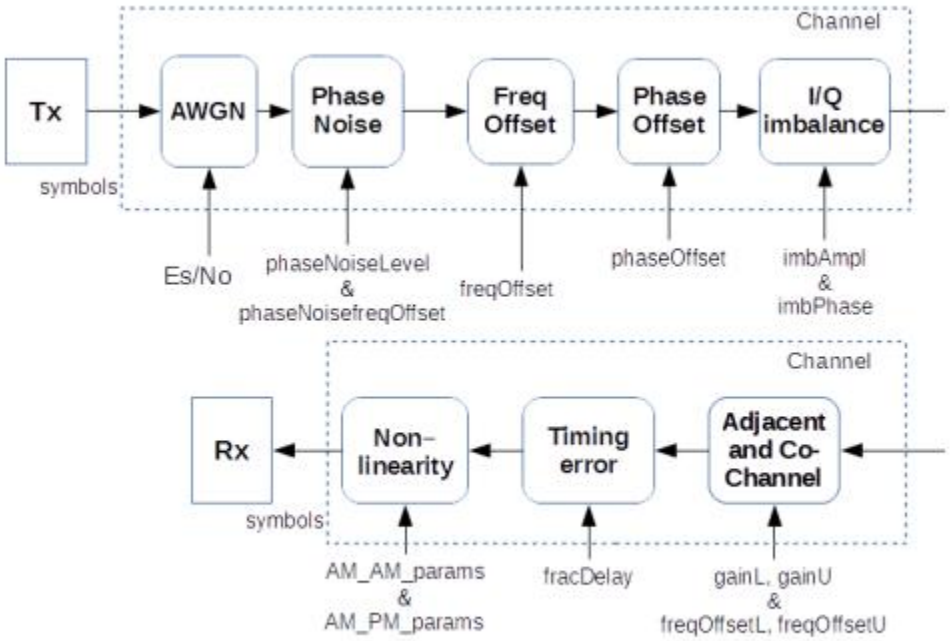

Figure 4. Channel simulation model

Communication Channel Simulation Physical layer impairments in the communication channel include those introduced by the channel, such as reflections and interference, as well as those induced by various components in the system, such as tuner I/Q imbalance and amplifier non-linearity. These impairments degrade the received SNR and may in some cases affect the convergence behavior of various computation loops in the receiver.

In order to test the demodulator performance, different realistic conditions that can affect the quality of received signals are simulated. Physical layer impairments in DVB-S2 receivers are discussed in [4]. A simpler channel model is implemented in Matlab (Figure 4). Every noise source is set independently, allowing flexible channel simulation. Receiver The functional block diagram of DVB-S2

Receiver

according to DVB-S2 guidelines [2] is depicted in Figure 6. The Receiver application includes the following functional blocks. Signal Processing Chain

- Adjacent Channel Filtering using BB FIR.
- I/Q imbalance compensation, an iterative algorithm to estimate I, Q and compensate for imbalance.
- DC offset removal, using a simple IIR.
- Frame Synchronization, using a 25 taps correlator and a peak detector.
- Symbol Timing Recovery, using a Farrow cubic interpolator and a Gardner detector.
- Decimator and Matched Filter.
- Carrier Frequency Recovery (coarse and fine recovery) based on a pilot. Coarse recovery employs a second order feedback loop based on a delay-and-multiply frequency error detector. Fine recovery employs a feed-forward (FF) estimation algorithm, derived from the L&R (Luise and Reggiannini) technique.

- Phase Recovery (coarse and fine recovery), using FF ML estimator.
- Digital AGC, based on a pilot assisted vector tracker mechanism.
- LMS Equalizer, employing DFE with a small number of taps. Decoder Chain

- LDPC Decoder, BCH Decoder, BB Descrambler and BB Header CRC Decoder.

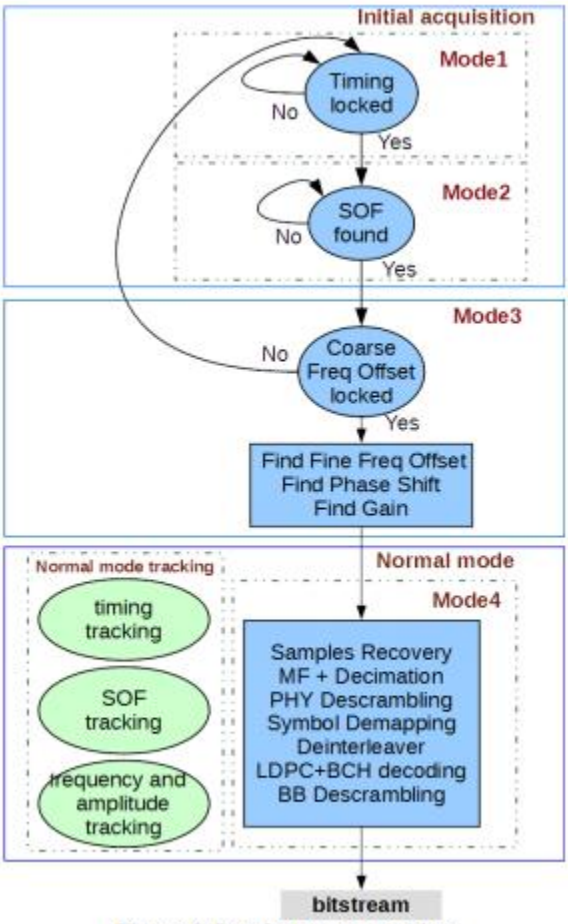

Figure 5. Receiver state machine

The performance of the DVB-S2/DVB-S2X link (consisting of transmitter, channel and receiver) is evaluated by the signal analyzer (Figure 2). The signal analyzer compares reconstructed bits with transmitted bits and calculates Frame Error Rate (FER), Packet Error Rate (PER) and Bit Error Rate (BER). In a communication chain without channel impairments, the reconstructed data should be

- Descrambler, identical to the TX scrambler
- LLR calculation, finding the logarithm of the distance between the soft symbol and the nearest hard
symbol.
- De-interleaver, identical to the TX interleaver.

exactly the same as transmitted. The DVB-S2 standard defines the expected error performance for different modes. PER is the ratio between the useful transport stream packets (188 bytes) correctly received and affected by errors, after forward error correction. Similar to the transmitter, the receiver, too, may be clustered into Pre-LDPC, LDPC and Post-LDPC stages. The RF Front End, ADC and AGC blocks are not implemented in the simulator. Figure 5 describes the state machine of the receiver. Steady-state is entered when acquisition stages complete successfully. The main computation during this state consists of filtering, PHY descrambling, de-mapping and de-interleaving. The FEC LDPC decoder is implemented as a hardware accelerator. The rest of the computation includes BCH decoding (in some cases), descrambling and header decoding. In parallel, tracking is performed for the next incoming frame, enabling fast reaction to channel impairment changes, modulation changes and end-of-stream detection

PERFORMANCE

This section reports performance results as computed with the RC64 DVB-S2 simulator and cycle-accurate simulations of RC64 [8].

Transmitter Performance

When simulating transmission of short frames using 32APSK modulation and LDPC code of 8/9, the Pre-LDPC stage requires 16,000 cycles, LDPC encoding takes 560 cycles, and Post-LDPC is 100,000 cycles. Since there are 3402 32APSK symbols in a short frame, Post-LDPC can be considered as incurring 30 cycles per symbol. As shown in Figure 11, a useful balance between pre-LDPC and post-LDPC can be achieved with nine frames per iteration for pre-LDPC, generating a total of

3402×9=30,618 symbols. Parallel processing of these symbols in Post-LDPC tasks is achieved by the remaining 55 cores. Each Post-LDPC task processes 30,618/55=557 symbols, taking 557×30=16,710 cycles. This schedule translates to a data rate of Each symbol contains two samples, and there are 6,804 samples per frame.

Receiver Performance

When receiving short frames in a steady state, the receiver spends 220,000 cycles in the Pre-LDPC stage, 4,000 cycles on average in the LDPC decoder, and 32,000 cycles in PostLDPC. The schedule of Figure 12 shows 8,000 cycles per iteration, receiving two frames per iteration, using 54 DSP

MODEM IMPLEMENTATION

Details of modem implementation are described in this section. We first discuss hardware accelerators, followed by data streaming, scheduling and mitigation of overhead.

CONCLUSIONS

We have described a high-performance implementation of DVB-S2 transmitter and receiver on RC64, predicted to exceed 2Gbps transmission and 1Gbps reception. A sixlevels development and simulation process has been described. Dynamic scheduling of tasks to cores, using the hardware scheduler and based on task oriented programming, resulted in a flexible solution that can easily be adapted to other modem parameters and other standards

REFERENCES

[1] Ran Ginosar and PelegAviely, RC64 – Many-Core Communication Processor for Space IP Router. In Proceedings of International Astronautical Conference, pp. IAC-15-B2.6.1. Jerusalem, Israel, Oct. 2015.

[2] DVB (2005). User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2). User guidelines ETSI TR 102 376 V1.1.1 (http://www.etsi.org).

[3] Morello, Alberto, and VittoriaMignone. "DVB-S2: The second generation standard for satellite broad-band services." Proceedings of the IEEE, vol. 94, no. 1, pp. 210-227. IEEE, 2006.

[4] Nemer, Elias. "Physical layer impairments in DVB-S2 receivers." In Second IEEE Consumer Communications and Networking Conference, CCNC, pp. 487-492. IEEE, 2005.

[5] Savvopoulos, Panayiotis, Nikolaos Papandreou, and ThAntonakopoulos. "Architecture and DSP Implementation of a DVB-S2 Baseband Demodulator." In Digital System Design, Architectures, Methods and Tools, 2009. DSD'09. 12th Euromicro Conference on, pp. 441-448. IEEE, 2009