

Design and Implementation of Variable-Rate QPSK Demodulator from Data Flow Representation

Seung-Jun Lee

Abstract

This paper describes the design of a variable rate QPSK demodulator for digital satellite TV system. This true variable-rate demodulator employs a unique architecture to realize an all digital synchronization and detection algorithm. Data-flow based design approach enabled a seamless transition from high level design optimization to physical layout. The demodulator has been integrated with Viterbi decoder, de-interleaver, and Reed-Solomon decoder to make a single chip Digital Video Broadcast (DVB) receiver. The receiver IC has been fabricated with a 0.5mm CMOS TLM process and proved fully functional in a real-world set-up.

I. Introduction

The Digital Video Broadcasting (DVB) standard for digital TV broadcasting by satellite was finalized in 1994 by European Broadcasting Union (EBU)[1]. It is the first open standard for digital Direct Broadcast Satellite(DBS) and has become the de facto world wide standard adopted by virtually every new satellite broadcast system. A primary objective of a satellite broadcasting system is to minimize the size and cost of the required receive antenna. QPSK modulation is optimum from the standpoint of power efficiency. In addition, a powerful forward error correction is specified which minimizes the total required received signal power at the home terminal. The forward error correction technique is a combination of convolutional coding with Reed-Solomon block coding which is referred to as concatenated coding. The functional block diagram of the entire encoding process at the transmitter is illustrated in Fig. 1.

The DVB standard provides for an MPEG-2 output data stream which is virtually error free to support video decompression. In addition to video, there are numerous other non-video satellite data services which will also conform to this standard. This flexibility is possible because the standard doesn't specify the transmission bit rate which remains to be

fixed by satellite operators and broadcasters. That implies the QPSK demodulator at the subscriber side should be able to handle a large variety of bit rates.

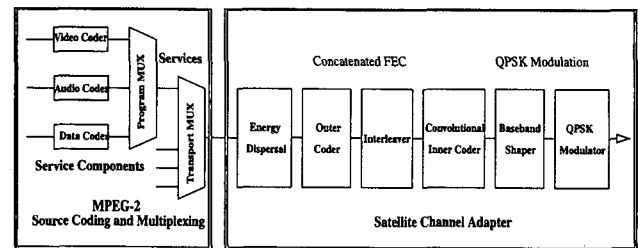


Fig. 1. DVB Specification

Following chapters will present the design and implementation of a variable-rate QPSK demodulator. This true variable rate receiver employs a unique architecture in which all operations are performed with a fixed sampling clock. How a data-flow based design tool helped to meet the design challenge in implementing such a complex digital system will be described in detail.

II. Digital Demodulator Algorithm and Architecture Design

Fig. 2 shows the configuration of the variable rate QPSK

demodulator. Unlike conventional synchronization schemes which include analog frequency tuning in the tracking loops, this product performs all tuning and filtering digitally. Thus the analog frequency translation can be performed with fixed frequency sources which have inherently low phase noise.

Baseband in-phase(I) and quadrature(Q) inputs are applied to the demodulator at a fixed sampling rate. The carrier frequency error associated with these samples is removed digitally through the decision-directed carrier recovery loop [2].

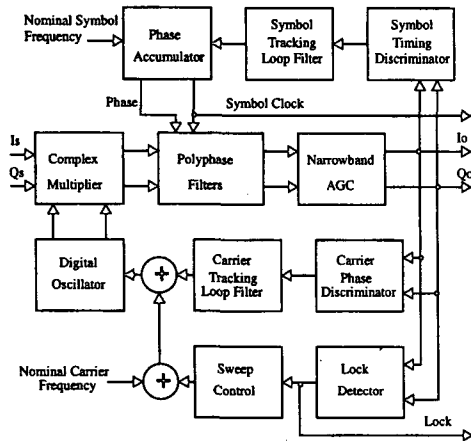


Fig. 2. Demodulator Block Diagram

Polyphase filters perform the square-root raised cosine filtering of the frequency-corrected baseband samples. These filters accept input samples and produce one output per received symbol[3]. They are always configured to have an impulse response duration of 4 symbols regardless of the programmed symbol rate. For low symbol rates, a large number of samples are used, while for high symbol rates a relatively low number of samples are processed for each filter output. The output of the polyphase filters are applied to a digital narrowband AGC which insures that the signal is optimally scaled to the Viterbi decoder to an accuracy of +/-0.5dB to insure optimum FEC performance.

A phase accumulator generates a clock to the filters at the symbol rate of the input signal. It is essentially a digital oscillator whose frequency is set to the nominal symbol rate. The symbol rate can be controlled within 0.01% of the desired value. Possible symbol timing errors are corrected through the symbol tracking PLL to maintain continuous phase lock with the received signal[4]. At each sample clock, the phase accumulator also produces a high resolution phase word which is used to control the instantaneous phase of the polyphase filters.

Fig. 3 compares the configuration of a demodulator system employing the full digital demodulator with the one using the conventional method. Without external analog PLL the new scheme provides much simpler system interface such that it

reduces the overall system cost.

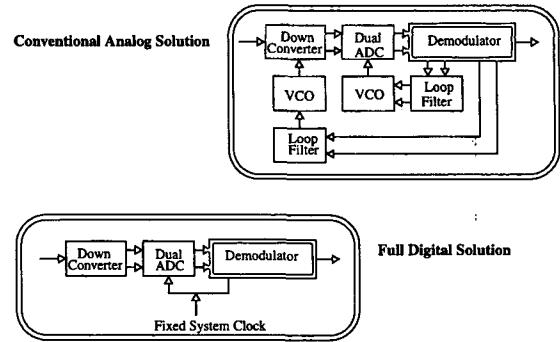


Fig. 3. System Interface Comparison

III. Data-Flow Based Design Methodology

The conventional design flow for a complex function, such as a QPSK demodulator would involve several phases. Typically, a C-code model of the algorithm would be developed using integer logic to simulate quantization effects. Once the algorithms had been tested, the design would be handed over to a logic designer who would translate C-code to gates. C-code modeling has the advantage of being readily available and easily simulatable. However, it was originally designed as a programming language with a single data stream processor in mind. Consequently it allows only single thread of control, which makes it hard to describe the modularized and concurrent behavior of a system.

Block diagram/equation type of notations may be the natural medium for a system designer. Therefore, data flow type descriptions tend to be favored for the description of high performance sub-systems and for the description of algorithms on a high level of abstraction. Several CAD tools are available to support system level design using data flow notation, among which are SPW[5], COSSAP[6], and Ptolemy[7]. They commonly provide the capability of building algorithmic level description and translating it into hardware, which can be summarized in three different scope.

1. Creating system schematic

They provide a graphical schematic editor to describe the target system. High-level modeling capability using C is usually supported for behavioral level description. Many building blocks are also provided in the form of library for rapid system proto-typing. Bottom-up design is possible by building hierarchical models using the basic library blocks.

2. Optimization by Simulation

They provide a stream-driven simulation engine to execute data-flow description. Various data sources and noise models are available to introduce noise, frequency offset, and inter-channel interference to the incoming signal. Graphical tools are also provided to monitor the performance in the forms of eye diagram, scatter diagram, histogram, etc. System model can be verified and optimized through extensive simulation in a various operating conditions.

3. HDL code generation

High level description won't be very useful unless it can be precisely translated to lower level description down to silicon level. To provide seamless translation to hardware, data flow based tools support automatic HDL code generation capability such that synthesizable HDL code is readily available from the schematic design and the behavior is preserved. That allows to get a gate-level description without no extra hooks and controls and the behavior is exactly the same as the data flow description of the design.

The demodulator block has many feed-back loops and complicated filter structure such that it can be best described using data-flow model. The data-flow representation of the demodulator has been optimized through extensive simulation, and then converted to HDL representation for automatic logic synthesis. The QPSK demodulator architecture and logic design was completed over a four month span by employing COSSAP, a high level system design tool from Synopsys. The main challenge in realizing these systems are designing the complex algorithms, the time-to-silicon, and the system level verification of the functionality of the implementation. COSSAP is an integrated design tool which enables the system designer to create logic, optimize algorithms and create test inputs to verify performance. Once the system functions have been optimized, there is a direct path to HDL generation, logic synthesis, and physical layout.

Fig. 4 is a COSSAP schematic which implements the digital demodulator shown in the previous figure. Bottom-up design approach has been used for direct HDL generation. The top level design has been built up hierarchically from primitive building blocks such as NAND, ADD, MUX, etc.

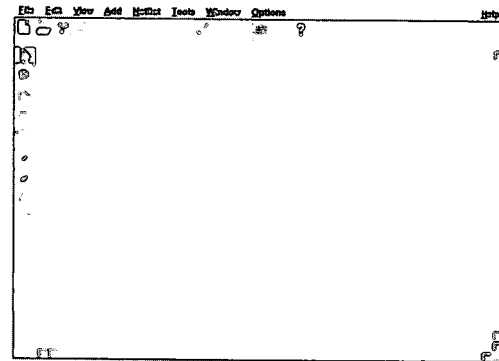


Fig. 5. Hierarchical Sub-Block

Fig. 5 shows one of the sub-blocks. Its function is to generate a control signal for the internal AGC block. A set of basic blocks make a more complex functional block.

The engineering challenge in the high speed digital signal processing functions is to implement algorithms which provide near theoretical performance despite quantization effects and technology limitations. Those data-flow based design tools are ideal for investigating, implementing and refining these constrained algorithms. Stream-driven simulation engine is efficient enough to try different design parameters in a reasonable amount of time. Fig. 6 illustrates carrier phase acquisition and tracking behavior at low SNR with different digital PLL loop gains. The performance of the carrier phase tracking loop can be fully characterized by varying the noise level, the initial frequency offset, and the gain parameters in the loop tracking filter.

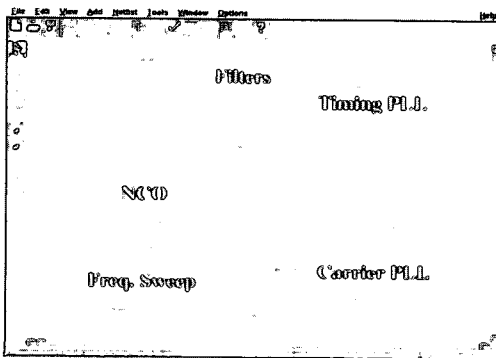


Fig. 4. Digital Demodulator Schematic

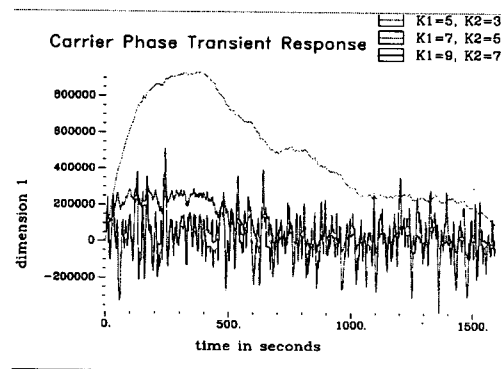


Fig. 6. Design Optimization by Simulation

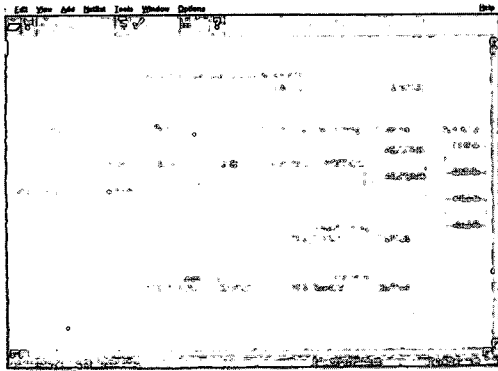


Fig. 7. Modeling of DVB Modulator

Fig. 7 represents the top level block diagram of DVB modulator. It can generate various test vectors to verify each sub-block of the receiver IC. The leftmost block is a DVB data source which provides FEC encoded data split into I/Q pairs. The digital I/Q data are filtered, combined with noise, filtered again, then sampled with simulated 6-bit A/D converters. Most of the models are provided in the form of library blocks such as filters and noise generators. Some models are generated from user-written C-like descriptions. The modeling of the convolutional encoder is shown in Fig. 8. It consists of three parts: input/output port definitions, initialization block which executes once in the beginning of the simulation, and the main body which executes whenever new data arrive at the input port.

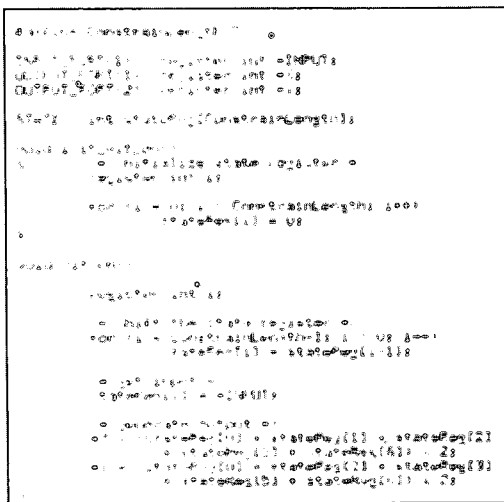


Fig. 8. Behavioral Modeling in COSSAP

IV. Digital Demodulator Implementation

In addition to evaluating the performance of the de-

modulator, it is also important to evaluate the complexity of the logic realizing the demodulator. Functions which are implemented with those basic building blocks can be converted directly to HDL such as Verilog or VHDL. The HDL code can be automatically synthesized into gate-level description. This process provides a measure of gate count (or area) as well as static timing analysis. In some cases pipelining must be added to the schematic and the process is repeated several times to achieve the desired speed goal. There are two important skills that can dramatically improve the area and the speed of the synthesized logic: manual description of timing-critical block and appropriate partitioning and flattening of the design.

The demodulator contains many multipliers and accumulators more than 20 bits, and it was not possible to meet the timing requirement at 60MHz operation starting from the generic multiplier and adder descriptions provided from the library. To meet the timing specification without modifying the original schematic, the HDL codes for several timing-critical arithmetic blocks have been replaced by manually-written codes which are known to generate faster logic. As a result, the synthesis could be successfully finished meeting all the timing requirements without adding additional pipeline delays, even though the area has been increased a little bit. Table 1 shows the speed improvement achieved by manual coding.

Table 1. Speed Improvement

BLOCK	Original	Modified
Adder (28 bits)	14.7 ns	9.4 ns
Adder (20 bits)	10.4 ns	8.56 ns
Subtractor (8 bits)	8.14 ns	6.9 ns
Multiplier (6 x 6)	13.2 ns	11.1 ns
Multiplier (6 x 7)	15.2 ns	12.3 ns

As the data-flow schematic was built in bottom-up fashion and the HDL code of the demodulator has been automatically generated from that schematic, it maintains a very deep and complicated hierarchy. There are generally two possible options in logic synthesis: hierarchical and flattened. In hierarchical synthesis, the whole design hierarchy is maintained such that resource-sharing is not allowed. Besides, it frequently happens that redundant buffers and inverters are inserted at the interfaces between sub-blocks. Consequently the total gate count increases significantly and the critical path delay may also get longer. When the design is flattened, the boundaries between sub-blocks are gone and the synthesis result improves in general. But the structural information is lost such that debugging process becomes difficult, and the synthesis takes too much time and the performance degrades when the size of the design gets bigger.

In the synthesis of the demodulator, the top level design

has been partitioned in such a way that both two options are used appropriately. Careful partitioning and flattening resulted in 18% reduction in critical path delay and 29% area reduction. Table 2 summarizes the synthesis results for hierarchical method, flattened method, and combined method.

Table 2. Synthesis Result with Different Options

Item	Hierarchical	Full Flattened	Combined
Area	42588 (129%)	42677 (129%)	33131 (100%)
Maximum Delay	18.06 ns (122%)	17.02 ns (115%)	14.86 ns (100%)

V. Chip Implementation

A single-chip DVB compliant receiver IC integrates the variable-rate QPSK demodulator with Viterbi decoder, de-interleaver, and Reed-Solomon decoder[8]. The top level block diagram of the receiver is shown in Fig. 9. The Viterbi decoder, de-interleaver, and Reed-Solomon decoder were described directly in HDL and automatically synthesized in hierarchical fashion.

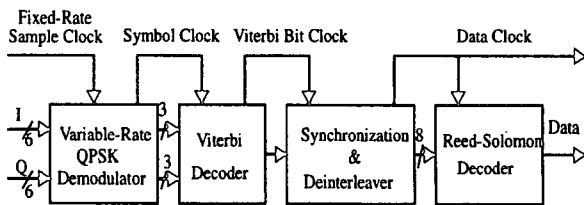


Fig. 9. DVB Receiver Block Diagram

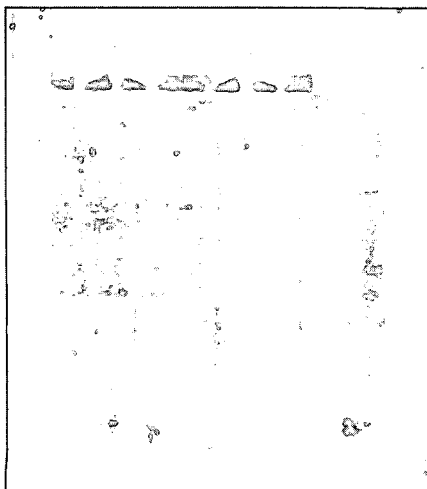


Fig. 10. Chip Photograph

To minimize the Silicon area the synthesized gate-level netlist has been flattened for place and route. All memory blocks have been manually laid out to minimize the size and to fit into the overall floorplan. Careful floorplan and flat place and route resulted in area-efficient layout with the size of 5.7x6.8mm² using 0.5mm CMOS TLM process, which is almost half of the previously reported result[9]. (Fig. 10).

The chip has been tested in a real-world situation with an 18 inch parabolic antenna as well as in a testing environment and proved to fully meet the DVB performance requirements.

VI. Conclusion

The design and implementation of a variable-rate QPSK demodulator has been presented. This true variable demodulator employs a unique architecture to realize an all digital synchronization and detection algorithm. The major challenge was to describe the complicated digital demodulator architecture efficiently and to translate the high level description to real hardware seamlessly. Data-flow based design approach enabled the architecture optimization, HDL code generation, and the test vector generation to be carried out uniformly in a single design environment. The demodulator has been integrated into a single DVB receiver IC with FEC blocks. The receiver IC has been fabricated with a 0.5mm CMOS TLM process. It has been extensively tested in a real-world set-up and proved fully functional

References

- [1] ETS 300 421 European Telecommunication Standard, "Digital Broadcasting Systems for Television, Framing Structure, Channels Coding and Modulation for 11/12 GHz Satellite Services", August 1994
- [2] E.A. Lee and D. G. Messerschmitt, "Digital Communication", 2nd Edition, KAP, Boston, 1994
- [3] US Patent 5,425,057 "Phase Demodulation Method and Apparatus Using Asynchronous Sampling Pulses", Michael Paff, June 13, 1995
- [4] F. M. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers", IEEE Transactions on Communications, vol. COM-34, pp. 423-429, May 1986
- [5] SPW User's Manual, Alta Group
- [6] COSSAP User's Manual, Synopsys
- [7] Almagest: Ptolemy User's Manual. Electronic Research Laboratory, University of California, Berkeley
- [8] S. Lee, J. Baek, et. al., "A Single Chip DVB Receiver for Variable-Rate QPSK Demodulation and Forward Error Correction", IEEE 1997 Custom Integrated Circuit Conference, Santa Clara, 1997

- [9] L. Christopher, J. Steward, et. al., "A Fully Integrated Digital Demodulation and Forward Error Correction IC for Digital Satellite Television", IEEE 1995 Custom Integrated Circuit Conference, Santa Clara, 1995



Seung-Jun Lee received the B.S. degree in electronics engineering from Seoul National University in 1986, and M.S. degree in electrical engineering from the University of California, Berkeley, in 1989, and Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1993.

He was joined Hyundai Electronics Industries in 1992, involved in the development of 16M DRAM, 256M Synchronous DRAM, and digital satellite TV receiver IC. and presently a staff development engineer engaged in efforts to develop various broadband modem products.