

AN ANALOG DECODER FOR (8,4) HAMMING CODE WITH SERIAL INPUT INTERFACE

Shuhuan Yu, Chris Winstead,
Chris Myers, Christian Schlegel, and Reid R. Harrison

Department of Electrical and Computer Engineering
University of Utah
Salt Lake City, Utah 84112
[ysh, harrison]@ece.utah.edu

ABSTRACT

A pipelined serial input interface is introduced for an analog BCJR decoder for an (8,4) Hamming code. The system uses arrays of sample-and-hold circuits to interface a continuous flow of serial channel data with a parallel, continuous-time analog decoder circuit. We fabricated this decoder in 0.5 μ m CMOS technology and verified its real-time operation decoding an 850kHz serial data stream with SNR above 5dB.

1. INTRODUCTION

Many coding techniques have been developed to accommodate the demand for efficient and reliable data transmission. Using error control coding, a communication system may achieve good error performance at low SNR and hence reduced transmitter energy.

Researchers have successfully used highly parallel continuous-time analog circuits to implement soft decision decoding algorithms for small codes [1][2]. High-performance error-correcting codes are quite large, and analog decoders require parallel analog inputs which must be held constant until the circuit converges. However, transmitted signals typically arrive serially from a channel. A serial-to-parallel analog interface is therefore necessary to make analog decoders a practical technology. This paper introduces a pipelined serial input interface that is implemented with an analog decoder for an (8,4) Hamming code. The Hamming code is briefly introduced in Section 2. The serial input interface and its sample-and-hold cell are described in Sections 3 and 4. Finally the experimental results from the fabricated chip are reported. The proposed circuit was built in a commercially-available 0.5 μ m CMOS technology.

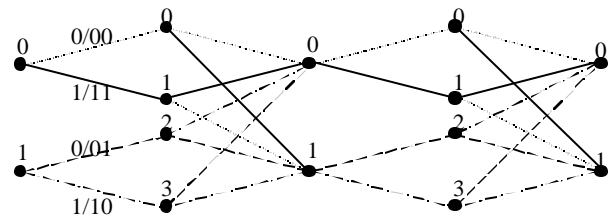


Figure 1: Minimal tail-biting trellis for Hamming code

2. BCJR DECODER FOR (8,4) HAMMING CODE

The Hamming code is a simple example of an error-correcting code, which adds redundancy to transmitted data so that errors introduced by a noisy communication channel can be reduced. From a block of four input information bits \underline{u} , the (8, 4) Hamming code encoder will generate eight output bits \underline{x} for transmission. The encoding operation is $\underline{x} = \underline{u}G$, where:

$$G = \begin{bmatrix} 11 & 11 & 00 & 00 \\ 00 & 11 & 01 & 10 \\ 00 & 00 & 11 & 11 \\ 01 & 10 & 00 & 11 \end{bmatrix} \quad (1)$$

The BCJR approach provides an optimal decoding algorithm [3]. As shown in [4], the decoder operates on the samples by propagating probability information through the minimal tail-biting trellis graph (Figure 1). The BCJR algorithm iterates around the graph and gathers both forward and backward estimates to generate the final estimates of the four information bits. The decoder's computation is mainly composed of sum-product operations. The computations in the decoder may be decomposed into basic cells which may be implemented in analog circuits constructed from bipolar or subthreshold CMOS transistors based on Gilbert multipliers [5].

In these analog, continuous-time decoding circuits, the inputs to a computation block must be provided simultaneously (i.e., both inputs and outputs for the

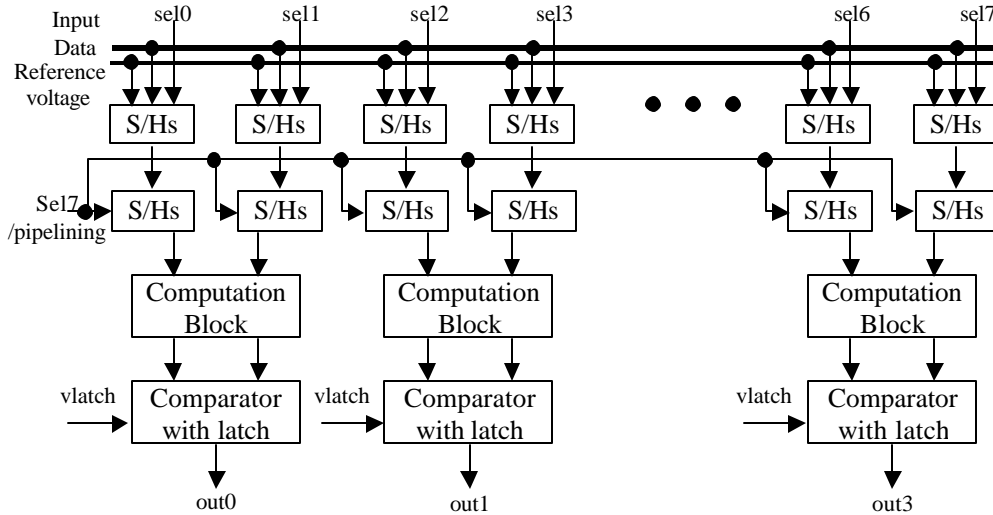


Figure 2: Architecture of the decoder chip.

computation block must be in parallel). The decoder for the (8,4) Hamming code receives 8 parallel input samples and produces 4 information bit estimates in parallel. Each analog input sample represents a “soft bit,” which is the log-likelihood ratio (LLR) as $\ln[P(1)/P(0)]$, based on a single channel measurement [6]. The simplest realization, of course, is to use parallel input and output channels for the chip, which is reported in [4]. However, pin limitations restrict this method to small codes. Also, data from a communications channel typically arrive serially, so a serial input interface is required to provide the computation core with parallel input signals.

3. THE SERIAL INPUT APPROACH

To realize a serial input with internally parallel computation, some researchers used an ADC at the input with an internal array of DACs to provide parallel analog signals [5]. The ADC/DAC realization adds complexity to the circuit design, consumes significant area and power, and introduces more digital noise into the chip. In this paper, we propose a mixed-signal pipelining plan to realize serial inputs with internally parallel analog circuits performing the error-control decoding. Without the power and area overhead consumed by data conversion, this analog realization should perform more efficiently.

Figure 2 shows a system block diagram for the decoder. There are two pipelining stages, each stage with 8 clock cycles, corresponding to the eight input bits. The design is fully differential to improve immunity from substrate noise and other extrinsic noise sources. There are two sample-and-hold (S/H) circuits for each “soft bit” to store both ends of a differential input voltage. One end of each sample is a common reference voltage. In the first stage, both the input data and the reference voltage are stored locally with sample-and-hold cells. At the second

stage, all the data are passed in parallel to the computation blocks for the analog computation when the pipeline signal is enabled. The “soft output” signals generated by the computation blocks represent the probabilities that the received bit was a “1” or a “0”. At the end of the second stage, comparators generate digital decisions by comparing the two probabilities and latch the data simultaneously for all the outputs. In the future, another stage can be added to send the data off chip serially by using shift registers.

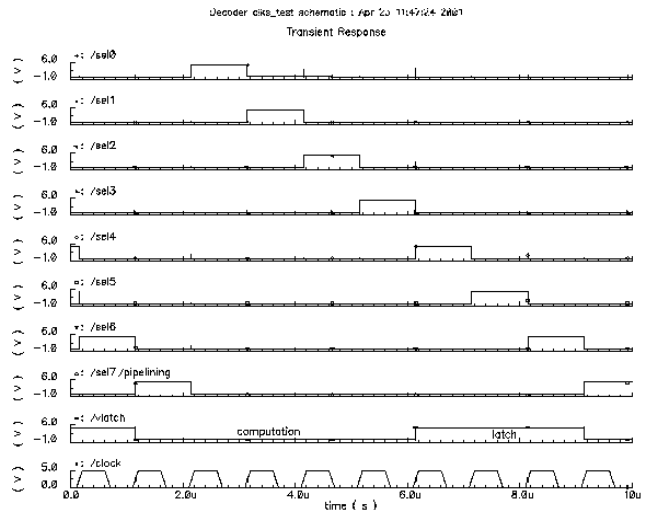


Figure 3: Timing diagram for analog decoder with pipelined input. “sel0” to “sel7” signals are the enabling signals for the S/H cells. “sel7” signal is also used for the pipeline-enabling signal. “v latch” signal is used in the comparator cells to trigger the comparators and synchronize the output signals at the second stage.

A detailed timing chart is shown in Figure 3. The “sel0” to “sel7” signals are the enabling signals for the S/H cells, which realize an analog MUX to store analog input

signals locally as the input data are scanned into the chip. The “sel7” signal is also used for the pipeline-enabling signal; all the data are passed in parallel to the computation blocks when it is enabled. The “vlatch” signal is used in the comparator cells to trigger the comparators and synchronize the output signals at the second stage. Most of the second stage clock cycle time can be used for the computation as the comparator only needs a few clock cycles to latch and settle down.

This input interface allows continuous serial data transmission without the need for any “breaks” or pauses in the input signal between code words. The circuit supplies parallel inputs for the core computation blocks while the serial input is running. This allows computation blocks to work N times slower than the serial input interface given an N -bit codeword. The computation blocks can work more slowly, giving the analog BCJR algorithm time to converge. Typically, the multiplication function cell (the Gilbert cell) [5] is working under moderate or weak inversion to minimize power consumption.

The input capacitance is dominated by the drain-to-substrate capacitance of the S/H switches and increases linearly with the number of input bits in the code. This may become a limiting factor for larger codes. A hybrid serial-parallel input scheme could be used if necessary.

4. SAMPLE AND HOLD CELL

The sample-and-hold cell is the key part for the serial input interface. The S/H cell in our chip is composed of one switch for selection, one capacitor for holding the data, and one buffer for isolating the signal if there is another stage following. A single input stage for the input data or reference voltage is illustrated in Figure 4.

There are several other advanced designs for S/H circuits [7]. Most of them try to minimize charge injection (clock feedthrough). This design uses the differential output of two S/H circuits. It is hoped that this differential design reduces the effect of charge injection. This serial input interface is designed for use with large codes, so area considerations preclude the use of a complex, area-consuming design.

The sizing for both p MOS switch and n MOS switch in this design is $W/L = 1.8\mu\text{m}/0.6\mu\text{m}$, and the storage capacitor has a value of 200fF. The speed of the S/H cells is calculated by:

$$R_{on}C = \frac{C}{g_{on-n} + g_{on-p}} = \frac{C}{C_{ox} \frac{W}{L} (m_n V_{effn} + m_p V_{effp})}$$

which is approximately 0.384ns in our technology with a 3.3V power supply. The clock period is required to be greater than $20R_{on}C = 7.7\text{ns}$ to ensure the capacitor has

enough time to charge, which means the S/H cell can work up to 200MHz.

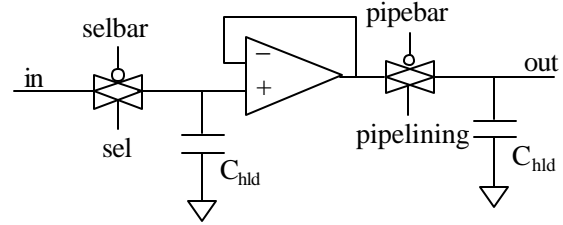


Figure 4: Single pipelined S/H input stage.

The charge injection and the leakage current of the S/H cells have been characterized in a fabricated test circuit. The leakage currents from p MOS and n MOS tend to cancel each other in a switch pair. The resulting total leakage current is quite small; the leakage current for a stored voltage between 0.5V and 2V (measured by observing the drift in the held value over a period of seconds) is in the range of $5.6 \times 10^{-17}\text{A}$ to $-1.1 \times 10^{-17}\text{A}$. A positive value means the leakage in the p MOS transistor is greater than for the n MOS and vice versa. The total leakage current is on the order of 10^{-17}A , meaning that the possible voltage change is only around 10nV even if the S/H cell operates as low as 1kHz.

When the S/H switch is turned off, the channel inversion charge is forced out from under its gate into both junctions, and the capacitor C_{hd} holds the charge injected at its side. If both p MOS and n MOS switches are on, and 50% of the channel charge is assumed to be injected into the hold capacitor, the voltage change is expressed as below:

$$\begin{aligned} \Delta V &= \frac{C'_{ox}WL}{2C_{hd}}(V_{effP} - V_{effN}) \\ &= \frac{C'_{ox}WL}{2C_{hd}}(2V_{in} + V_{tp} + V_{tm} - V_{dd}) \end{aligned} \quad (3)$$

Charge injection measurements have been taken for five chips. The results show that the voltage shift caused by the charge injection on a single channel is no more than 24mV. As predicted by (3), the measured charge injection was minimized for inputs near $V_{dd}/2$. With the same amount of charge flow from the junctions, the voltage change caused by the charge injection can be reduced by increasing the value of the hold capacitor. However, this comes at the expense of speed. These results suggest that charge injection is the dominant factor limiting the accuracy of our input circuit.

5. TEST RESULTS

The circuit described above has been fabricated in a commercially-available $0.5\mu\text{m}$ CMOS technology. As shown in Figure 5, two programmable arbitrary waveform

generators (Agilent 33120A) are used to produce the input data signal and the clock signal. Those two signals are synchronized, and the synchronization signal is also used to produce the reset signal for the chip. The four-bit output codes are recorded by a mixed-signal oscilloscope (Agilent 54622D), which is capable of simultaneously monitoring all relevant analog and digital signals. All test instruments are controlled and monitored by a computer over a GPIB bus.

According to the principle of the encoding/decoding algorithm, a frame of 106 4-bit source messages is generated randomly. These 4-bit messages are encoded into 8-bit Hamming code words using (1). Noise is added to generate a simulated channel output with a specified SNR. These noisy, coded data waveforms are programmed into the arbitrary waveform generator to provide the serial input signal into the chip at a rate of 425kHz. Figure 6(a) shows one such analog serial input signal to the chip. Figure 6(b) shows the system clock and the four digital output bits. Note that the parallel decoded output signals are eight times slower than the serial input bits. Decoding occurs continuously, and the pipelined input design allows computation to proceed at a constant flow with no need to pause the input signal after every eight bits.

There is a design flaw in the comparator of the last output bit Dout3, so we used an off-chip comparator instead. This led to glitches and a delay on this channel. Those errors are corrected in post-processing.

For SNR values above 5dB, the test results show that there is no error for any frame of 106 codes. We repeated the experiment with an 850kHz input rate with identical results. We are currently working towards testing the system at higher speeds. Because we are expecting error rates below 10^{-5} for this code, we still need to provide larger test vectors to properly characterize its performance. However, the preliminary results demonstrate that the serial pipelined analog input interface works as designed.

REFERENCES

- [1] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI", *IEEE Trans. On Information Theory*, 47:2:837-843, Feb. 2001.
- [2] J. Hagenauer, M. Morez, and E. Offer, "Recent progress in decoding with analog VLSI networks," *2000 Conference on Information Sciences and Systems*, Princeton University, March, 2000.
- [3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, 20:284-287, March 1974

- [4] C. Winstead, J. Dai, W. J. Kim, S. Little, Y.-B. Kim, C. Myers, and C. Schlegel, "Analog MAP Decoder for (8, 4) Hamming Code in Subthreshold CMOS," *Proc. Advanced Research in VLSI*, Salt Lake City, pp. 132-147, March 2001.

- [5] F. Lustenberger, "On the design of analog VLSI iterative decoders," Ph.D. Thesis, Swiss Federal Institute of Technology, 2000.

- [6] M. Morez, J. Hagenauer, and E. Offer, "On the analog implementation of the APP (BCJR) algorithm," *ISIT2000*, Sarrento, Italy, June 25-30, 2000.

- [7] D.A. Johns and K. Martin, *Analog Integrated Circuit Design*, Wiley, 1996.

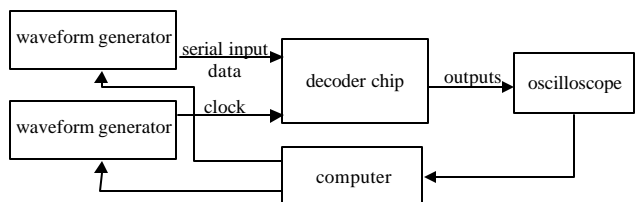


Figure 5: Test equipment setup

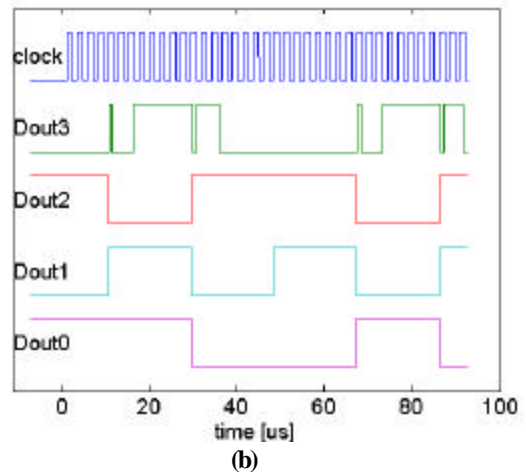
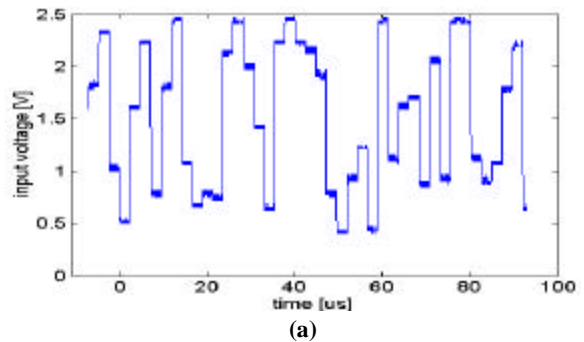


Figure 6: (a) Analog input signal and (b) digital signals, grabbed from oscilloscope. Output signals are 8 times slower than the clock with 8 serial input bits.