500 Mb/s Soft Output Viterbi Decoder

Engling Yeo Stephanie Augsburger, Wm. Rhett Davis, Borivoje Nikolić
University of California, Berkeley, CA 94720
{yeo, stephann, wrdavis, bora}@eecs.berkeley.edu

Abstract

Two 8-state, 7-bit soft output Viterbi decoders matched to an EPR4 channel and a rate-8/9 convolutional code are implemented in 0.18μm CMOS technology. Architectural transformation of the add-compare-select structures and modification of the register exchange allow a high throughput with small area overhead. The 4mm^2 chip has been verified to decode at 500Mb/s with 1.8V supply. These decoders are used with Turbo codes, which have been demonstrated to achieve information rates very close to the Shannon limit.

1. Introduction

This work demonstrates the implementation of two decoders applying the Soft Output Viterbi Algorithm (SOVA) [1]. These decoders can be employed as soft-input-soft-output (SISO) decoders for a Turbo coded system.

Figure. 1 shows an example that comprises a serial concatenation of an 8-state (11,13) convolutional encoder, with an enhanced partial response class-4 (EPR4) channel.

In order to achieve desired throughputs that are in line with current trends in magnetic recording systems, a fully unrolled and pipelined architecture [2] is needed. This results in a linear complexity increase, and will limit the number of iterations in practical systems to about three or four. Each SOVA decoder outputs 7-bit sign-magnitude values that consist of one decoded bit (hard output) and a 6-bit unsigned log-likelihood ratio (soft output). The soft output is based on the difference in path metric between the two most-likely (ML) paths, α and β, that trace back to complementary bit decisions, $\hat{x}$ and $\bar{\hat{x}}$, as shown in Figure 2. Both decoders have the same architecture, but are matched to different generator polynomials. The inner and outer decoders are named SOVA_EPR4 and SOVA_11_13 respectively to indicate the particular type of convolutional codes that each is used to decode.

The arithmetic computation and system architecture of the SOVA decoder will be discussed in Section 2. Section 3 highlights the micro-architectural analysis of the add-compare-select (ACS) structures, which are the traditionally speed-bottleneck of Viterbi decoder designs. Continuing the emphasis on high throughput rates, section 4 describes the use of deeply pipelined mechanisms for the traceback, equivalence detecting, and comparison of competing path metrics. Finally, Section 5 discusses the design flow as well as testing methodologies and results.

2. SOVA Decoder Architecture

The SOVA decoder outputs the log-likelihood of a correctly decoded bit. This value is given by the difference between the path metrics of the two most-likely (ML) paths that trace back to complementary bit decisions, $\hat{x}$ and $\bar{\hat{x}}$. Figure 2 shows that the ML path, α, is determined using the Viterbi algorithm with an L-step traceback. This is followed by another M-step traceback that resolves the next ML path, β, based on maximal probability of a deviation from α.

It is assumed that the absolute values of the path metrics, M_α and M_β, dominate over that of other paths, such that the probability of selecting β over α (i.e. the wrong decision) is given by (1). The log-likelihood of a correct output by the SOVA decoder is given in (2).

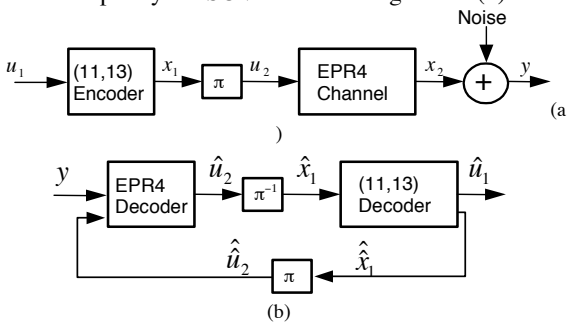

Figure. 1 Serial Turbo (a) encoder / (b) decoder with blocks separated by interleavers/ deinterleavers (π /π^{-1}).

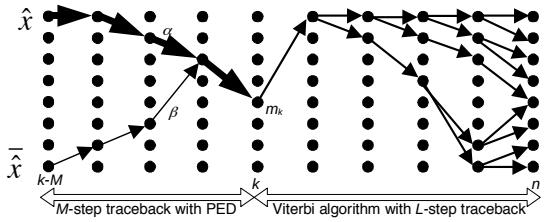

Figure 2. Two-stage traceback in a SOVA decoder to determine the two ML paths, α and β.

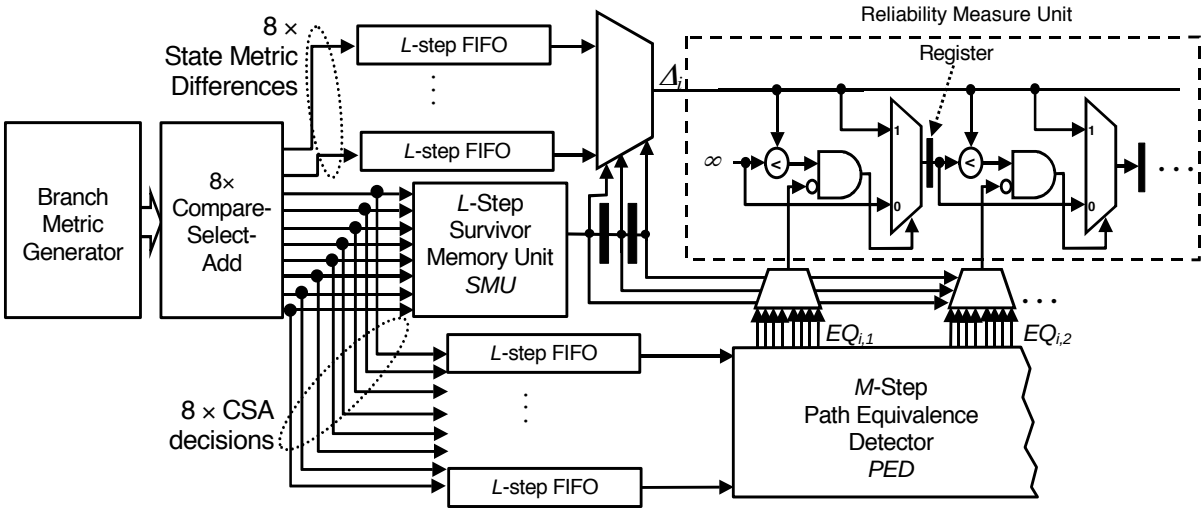

Figure 3. System architecture of 8-state SOVA decoder.

$$P_{err} = \frac{\exp(-M_\beta)}{\exp(-M_\alpha)+\exp(-M_\beta)} = \frac{1}{1+\exp(\Delta)} \quad ; \quad \Delta = M_\beta - M_\alpha \tag{1}$$

$$\log\left[P\left(\frac{CorrectDecision}{WrongDecision}\right)\right] = \log\left(\frac{1-P_{err}}{P_{err}}\right) = \Delta = M_\beta - M_\alpha \tag{2}$$

The system architecture of this implementation is shown in Figure 3. The branch metric generator, compare-select-add (CSA), and *L*-step survivor memory unit (SMU) form the building blocks of a conventional Viterbi decoder. The CSA is a retimed and transformed version of the more common add-compare-select (ACS) structure, and provides higher throughput rate at a lower area cost than traditional loop unrolling methods [5] [6] [8]. There are eight copies of the CSA in accordance to the targeted eight-state convolutional codes.

In addition to providing a path decision at each iteration, the CSA of the SOVA decoder is also required to output the difference in path metric between the two ML paths. The path decisions and the metric differences are cached into an array of *L*-step FIFO buffers. The delayed signals are used in the *M*-step path-equivalence detector (PED) to determine the similarity between each pair of competing decisions obtained through a *j*-step traceback, $j \in \{1,2,...,M\}$.

Finally, the output decisions from the SMU are used to select the delayed metric difference and the equivalence outputs corresponding to the most-likely state. These signals are input to a reliability measure unit (RMU), which outputs the minimum path metric difference reflecting complementary bit decisions, $\hat{x}$ and $\bar{\hat{x}}$.

3. Add-Compare-Select Structures

The throughputs of SOVA decoders have traditionally been limited by the implementation of the add-compare-select (ACS) structure due to a single-step recursion that prevents pipelining.

Previous high throughput implementations of the Viterbi decoder, [5] [6] [8], resorted to unrolling of the ACS loop in order to achieve high throughputs. These methods increase the critical path delay, but improve the overall throughput. However, in a soft-output Viterbi decoder implementation, additional overhead includes the modifications to the register-exchange and reliability measure units in order to handle the doubled symbol rate. These considerations render loop-unrolling an unsuitable technique.

In this design, the critical path constraint was eased through a process of retiming and transformation of the ACS recursions. The sequence of operations in a single stage of the pipeline is reordered as compare-select-add (CSA). This is followed by a transformation [3] [4] that moves the add operation ahead of the select operation, such that the compare and add operations are executed in parallel. This modification decreases the critical path delay at the cost of doubling the number of adders and multiplexers.

The adders in the transformed CSAs have flat input data-arrival profiles, and permit datapath synthesis to achieve the fastest adder implementation. The critical delay of the resulting structure (Figure 4) is reduced by 42% at the expense of 22% increase in overall area of the decoder.

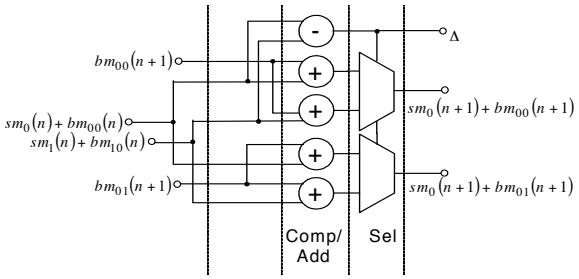

Figure 4. Transformed compare-select-add (CSA) structure.

4. SMU, PED and RMU structures

The two ML paths are determined by a dual-traceback function, achieved by cascading the SMU with a combination of the PED and RMU.

The SMU is implemented with a high-speed pipelined register exchange, shown in Figure 5. It outputs a decision relating the most likely state $\hat{\imath}(n)$ after a latency of L-cycles. This method avoids the complexity of designing specialized SRAM blocks with complex read/write control mechanisms, which was required by previous implementations, [5], [7]. It has been shown in studies of the Viterbi decoder that SRAM-based traceback has a costly throughput overhead due to the need to access multiple memory pointers [8]. The larger memory blocks required for storage of the 7-bit soft metrics, as opposed to single-bit decisions in the Viterbi decoder, resulted in further delays for memory access. In addition, the retiming and transformation of the ACS unit, traditionally the speed bottleneck of Viterbi decoder designs, has improved the critical delay by 42%. This imposes a more stringent throughput requirement on the traceback operation.

The PED is a modified register exchange (Figure 6) that provides Boolean outputs, $EQ_{i,j}(n)$ for j = 1, 2, .., M, indicating the equivalence between the two competing decisions obtained through a j-step traceback from state i.

From CSA_i, the difference between the two path metrics, $\Delta_i(n)$, arriving at time n, state i, is retained; $i \in \{1,2,...,8\}$. The output from the SMU selects $\Delta_i(n)$ and $EQ_{i,j}(n)$, which correspond to the values along the ML path, as inputs to the RMU.

The RMU consists of comparators and multiplexers in a pipeline that selects the minimum $\Delta(n)$ along the ML path. It is initialized with the maximum possible reliability measure represented as "∞" in Figure 3. Based on the EQ input, each pipelined section outputs one of the following:

EQ = 1: Reliability measure from the previous step
EQ = 0: Min{Δ_i, previous reliability measure}

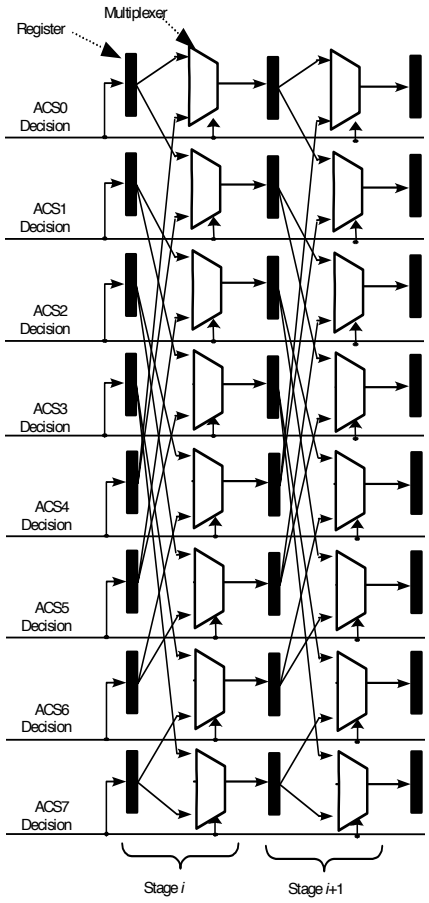

Figure 5. Example 8-state register exchange survivor memory unit used in VA-SMU.

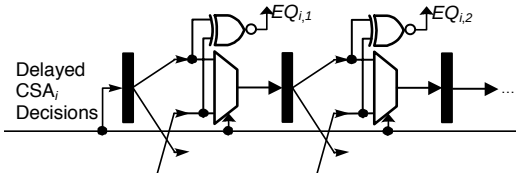

Figure 6. State-slice of register exchange used in the path-equivalence detector (PED).

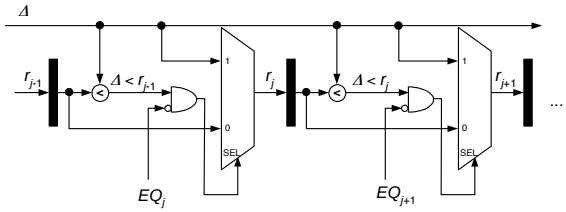

Figure 7. Pipelined section of reliability measure unit (RMU).

Compared with a Viterbi decoder implementation, the total size of the SMUs is approximately doubled (L = M). The RMU overhead consists of M copies of 1 register, 2 multiplexers and a 2-input comparator. The latency through the SOVA decoder is $L + M$. Both

decoder implementations use $L = M = 15$, which is five times the constraint length of the convolutional code. The additional latency remains insignificant compared to the overall latency in the Turbo-SOVA system, which is dominated by the latency through the interleavers.

5. Design Flow and Testing

The design of the SOVA decoders uses an automated design flow for direct-mapping of signal processing algorithms into integrated circuits [8]. This automated flow was further enhanced for high-speed design through customization of the clock tree to achieve low clock skews.

The functionality of the chip has been verified with 1.8V supply at 25°C. Throughput rates above 500 Mb/s were achieved and power dissipation was 400mW. The speed characterization was performed using a clock tree with a built-in delay line. The speed and power performance for one of the SOVA decoders is plotted in Figure 9. The power measurements were performed at the highest frequencies permitted by the supply voltage. Table 1 summarizes the characteristics of the decoders.

6. Conclusion

The design of a 500MHz soft-output Viterbi decoder has been described. It can be employed as a soft-input-soft-output (SISO) decoder for Turbo code systems. Architectural transformation of the add-compare-select structures and modification of the register exchange allow a high throughput with small area overhead.

In addition to magnetic recording applications, the SOVA decoder is also appropriate for Turbo-coded forward error correction applications in wireless, wireline, and optical communication systems.

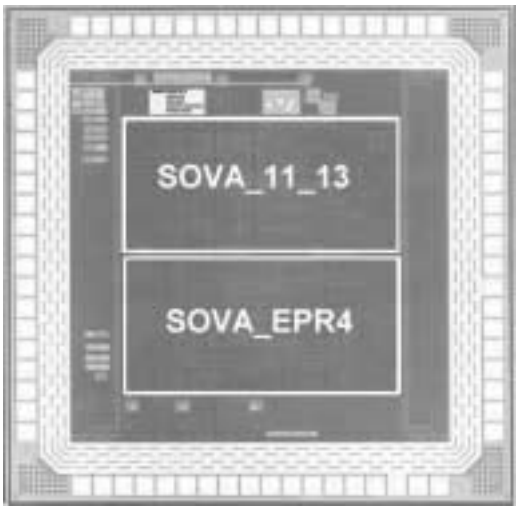

Figure 8. Die micrograph.

Table 1. Summary of results

Decoder Type	SOVA_EPR4	SOVA_11_13
Number of States	8	8
Transistor Count	164K	174K
Core Area	1mm × 0.5mm	1mm × 0.5mm
Speed	500Mb/s	500Mb/s
Avg. Power @ 400MHz	395mW	400mW

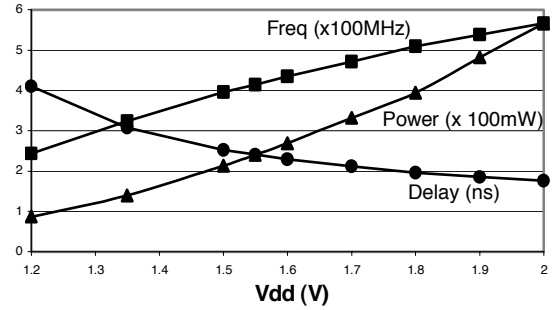

Figure 9. Performance of SOVA_EPR4 decoder.

7. Acknowledgements

The authors thank T. Smilkstein, P. Pakzad and V. Anantharam of UC Berkeley and B. Wilson of IBM for their technical assistance, and ST Microelectronics for fabrication of the test chip. Texas Instruments supported this work under the UC MICRO program.

8. References

[1] J. Hagenauer and P. Hoecher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proc. IEEE GLOBECOM*, pp. 47.11-47.17, Dallas, TX, Nov 1989.

[2] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Trans. Magnetics*, vol.37, no.2, pp.748-55, Mar. 2001.

[3] G Fettweis, R. Karabed, P.H. Siegel, and H.K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signaling," *Proc. IEEE GLOBECOM*, pp.559-63, Singapore, Nov 1995.

[4] I. Lee and J. L. Sonntag, "A new architecture for the fast Viterbi algorithm," *Proc. IEEE GLOBECOM*, pp.1664-8, San Francisco, CA, Nov. 2000.

[5] O. J. Joeressen and H. Meyr, "A 40 Mb/s soft-output Viterbi decoder," *IEEE JSSC*, vol.30, no.7, pp.812-18, Jul. 1995.

[6] Yeung, and J. M. Rabaey, "A 210 Mb/s radix-4 bit-level pipelined Viterbi decoder," Proc. *IEEE ISSCC*, San Francisco, CA, USA, pp.88-9, 344, 440, Feb 1995.

[7] D. Garrett and M. Stan, "Low power architecture of the soft-output Viterbi algorithm", *Proc. IEEE ISLPED*, Monterey, CA, pp.262-7, Aug. 1998.

[8] P. Black and T. Meng, "A 1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE Journ. Of Solid-States Circuits*, vol.32, no.6, pp.797-805, Jun 1997.

[9] W. Rhett Davis, Ning Zhang, Kevin Camera, Dejan Markovic, Tina Smilkstein, M. Josie Ammer, Engling Yeo, Stephanie Augsburger, Borivoje Nikolic, Robert W. Brodersen, "An Automated Design Flow for High-Throughput Low-Power Dedicated Signal Processing Systems," *IEEE JSSCC*, vol. 37, no. 3, pp.420-431, Mar 2002.