A 500-Mb/s Soft-Output Viterbi Decoder

Engling Yeo, *Student Member, IEEE*, Stephanie A. Augsburger, *Student Member, IEEE*, W. Rhett Davis, *Member, IEEE*, and Borivoje Nikolić, *Member, IEEE*

***Abstract*—Two eight-state 7-bit soft-output Viterbi decoders matched to an EPR4 channel and a rate-8/9 convolutional code are implemented in a 0.18-μm CMOS technology. The throughput of the decoders is increased through architectural transformation of the add-compare-select recursion, with a small area overhead. The survivor-path decoding logic of a conventional Viterbi decoder register exchange is adapted to detect the two most likely paths. The 4-mm^2 chip has been verified to decode at 500 Mb/s with 1.8-V supply. These decoders can be used as constituent decoders for Turbo codes in high-performance applications requiring information rates that are very close to the Shannon limit.**

***Index Terms*—Iterative decoders, turbo codes, Viterbi decoder, VLSI.**

I. Introduction

TURBO CODES consist of two or more convolutional codes concatenated through an interleaver in a parallel or serial structure. These codes are decoded iteratively by passing *a posteriori* probabilities between the decoding modules, and are capable of operating near the Shannon limit of channel capacity. There has been a recent interest in using iterative decoding techniques in advanced communication systems as well as magnetic disk-drive read-write channels [1]. Contemporary magnetic recording systems are based on partial response equalization with maximum likelihood detection (PRML) [2]. Sustained growth in the areal densities beyond 100 Gb/in^2 in future systems will be accompanied by reduced signal-to-noise ratios (SNRs) at which data is detected, requiring a shift toward more sophisticated detection methods such as iterative decoding. The application of iterative decoding [3] offers large SNR advantage over conventionally used PRML systems, but requires a significant increase in computational complexity. Magnetic storage channels differ from most other communications channels because they have very high throughput requirements under low cost and low power constraints.

The magnetic recording channel can be used as an inner, rate-1 encoder [1] for a serial Turbo code. Fig. 1 shows an example system that comprises a serial concatenation of an eight-state Octal(13) convolutional encoder, with an enhanced partial response class-4 (EPR4) channel. EPR4 equalization matches well with the channel response at contemporary recording densities. In addition to decoding incoming bits, the two decoders have to provide soft output information as a measure of the *a posteriori* probability. This computation is frequently performed using maximum *a posteriori* (MAP) decoders that implement the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [4]. The complexity of MAP decoders can be traded for marginally degraded bit error rate (BER) performance by replacing them with decoders applying the soft-output Viterbi algorithm (SOVA) [5]–[7].

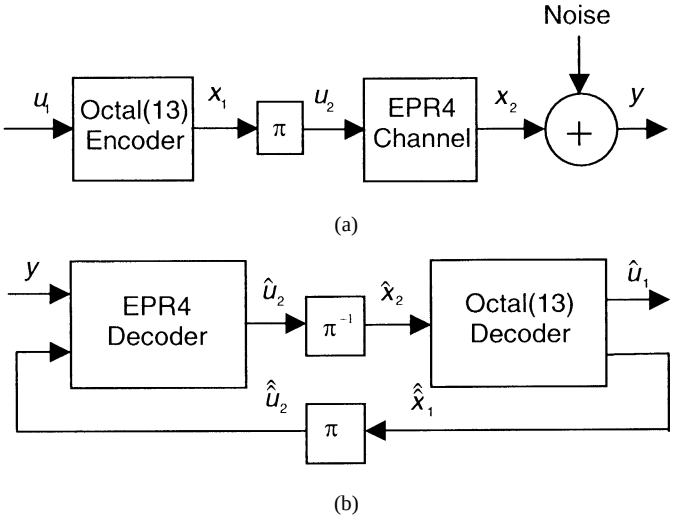


Fig. 1. Serial turbo. (a) Encoder. (b) Decoder with blocks separated by interleavers/ deinterleavers (π/π^{-1}).

In order to achieve throughputs that are in line with current trends in magnetic recording systems, a fully unrolled and pipelined architecture [6] is needed. This results in a linear complexity increase with the number of iterations, and therefore limits the number of iterations in practical systems to about three or four. The large area and power consumption of unrolled iterative decoders are two of the major challenges for their acceptance in storage systems.

An early VLSI implementation of a SOVA decoder [8] achieved 40 Mb/s throughput in a 1-μm CMOS standard cell technology. In order to reduce the power and area of the implementation, RAM macros were used. The path selections were done with the register-exchange technique to reduce the overall latency. A low power implementation of the SOVA decoder [9] uses DRAM blocks for path selection. The DRAMs need to be clocked at a multiple frequency of the decoding symbol rate. This places the memory as the limiting factor in the decoding throughput of the design.

Manuscript received December 1, 2002; revised March 15, 2003. This work was supported by Texas Instruments Incorporated under a UC Micro Program.

E. Yeo and B. Nikolić are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: yeo@eecs.berkeley.edu; bora@eecs.berkeley.edu).

S. A. Augsburger was with the University of California, Berkeley, CA 94720 USA. She is now with Intel Corporation, Santa Clara, CA 95052 USA (e-mail: stephanie.a.augsburger@intel.com).

W. R. Davis was with the University of California, Berkeley, CA 94720 USA. He is now with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA (e-mail: rhett_davis@ncsu.edu).

Digital Object Identifier 10.1109/JSSC.2003.813250

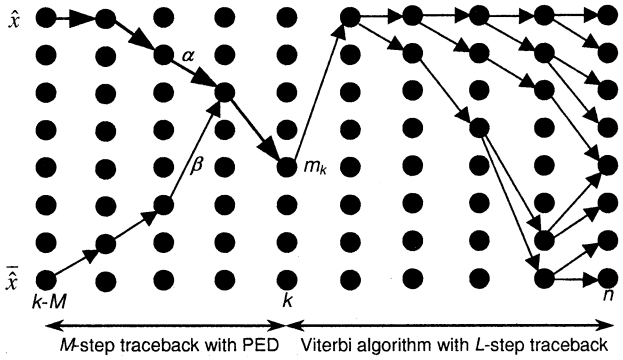


Fig. 2. Two-stage traceback in a SOVA decoder to determine the two ML paths, α and β.

In contrast, this work increases the throughputs of two implemented soft-output Viterbi decoders by means of retiming followed by transformation of the add-compare-select (ACS) recursion. These steps permit the computations of the add and compare steps to be performed in parallel. The path selection mechanisms make use of a modified register-exchange [8] to avoid the throughput bottleneck in SRAM blocks.

The arithmetic requirements and system architecture of the SOVA decoder will be discussed in Section II. Both decoders share the same architecture. One decoder is matched to the EPR4 channel with a $(1 \oplus D)^{-1}$ precoder while the other is matched to an Octal(13) feedforward convolutional code.

Section III presents the microarchitectural analysis of various ACS structures in power, area, and delay space. Section IV describes the use of deeply pipelined mechanisms for the traceback, equivalence detection, and comparison of competing path metrics. Finally, Section V discusses the design flow, test procedures, and results of tests performed on the decoder chip.

II. 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 $\bar{\hat{x}}$. Fig. 2 shows that the ML path, α, is determined using the Viterbi algorithm with an L-step traceback. This is followed by an M-step traceback that resolves the next ML path, β, based on maximal probability of its deviation from α.

Assuming that the absolute values of the path metrics, M_α and M_β, dominate over those of other paths, the probability of selecting β over α (i.e., making the wrong decision) is given by

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

The log-likelihood of a correct output by the SOVA decoder is given by

$$\log\left[P\left(\frac{\text{Correct Decision}}{\text{Wrong Decision}}\right)\right] = \log\left(\frac{1-P_{\text{err}}}{P_{\text{err}}}\right) = \Delta = M_\beta - M_\alpha. \tag{2}$$

Each implemented SOVA decoder outputs 7-bit sign-magnitude values. The sign bit carries the decoded bit decision (hard output), which is the same information output by a traditional Viterbi decoder. The 6-bit magnitude value represents the log-likelihood of an error in decoding of the particular bit (soft output).

The architecture that implements this decoder is shown in Fig. 3. The branch metric generator, eight compare-select-add (CSA) units, and the L-step survivor memory unit (SMU) form the building blocks of a conventional Viterbi decoder. The eight parallel CSA blocks compute the pairs of cumulative path metrics and select the winning paths in the underlying trellis representation of the convolutional code. Additionally, each CSA also outputs the difference in path metrics between the two competing paths. The path decisions are stored into an array of L-step flip-flop-based FIFO buffers. The delayed signals are used in the M-step path-equivalence detector (PED) to determine the equivalence between each pair of competing decisions obtained through a j-step traceback, $j \in \{1, 2, \ldots, M\}$.

The path metric differences from the eight CSAs are stored in FIFO buffers of depth L. Using the output decision from the SMU as a multiplexer select signal, the delayed metric difference at the most likely state is sent to a reliability measure unit (RMU). The RMU also receives a list of equivalence test results that are performed on the competing traceback decisions paths that originate from the most likely state. The selected equivalence results are evaluated in the RMU in order to output the minimum path metric difference reflecting competing traceback paths that result in complementary bit decisions, $\hat{x}$ and $\bar{\hat{x}}$.

III. ACS Structures

The throughput of hard- or soft-output Viterbi decoders is set by the particular target application requirements. Depending on the implementation platform or the complexity limitations, the decoders can be built using concurrent computation of all state metrics or by resource sharing through multiplexing the computational units [10]. High-throughput applications require the use of fully parallel decoder implementations.

The throughput of a SOVA decoder has traditionally been limited by the difficulty of pipelining the single-step ACS recursion. Fig. 4 shows the transition trellis of an example eight-state hard or soft-decision Viterbi decoder. The critical-path of a traditional ACS computation extends through the sequential execution of two parallel additions, a comparison and a selection. Let $sm_i(n)$ represent the path metric for state i, and $bm_{ij}(n)$, the branch metric of a corresponding transition from state i to state j, with the time step denoted by n. Then, an example of the ACS recursion corresponding to state 0 is shown in (3).

$$sm_0(n+1) = \min\left\{\begin{array}{l} sm_0(n) + bm_{00}(n) \\ sm_4(n) + bm_{40}(n) \end{array}\right\}. \tag{3}$$

The comparison is implemented through subtraction, and the most significant bit (MSB) of the result selects the winning path. The ripple-carry implementations of both add and compare operations take advantage of the similarity in carry profiles. The amount of overhead in the critical path required for executing the subtraction only involves the computation of the MSB of the difference. Fast adder structures such as the carry-select adder will require the subsequent subtraction to follow an abrupt carry profile, which yields minimal performance gains

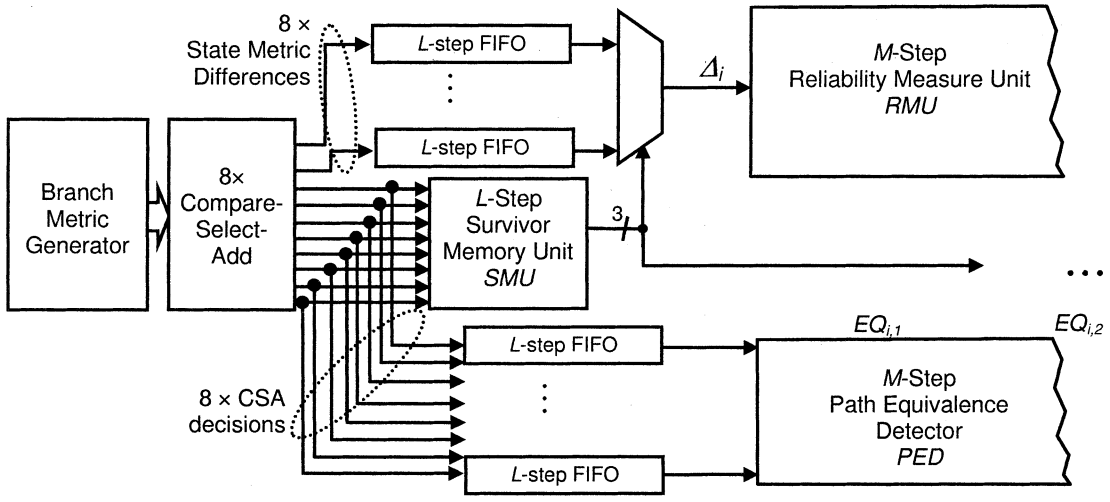


Fig. 3. System architecture of eight-state SOVA decoder.

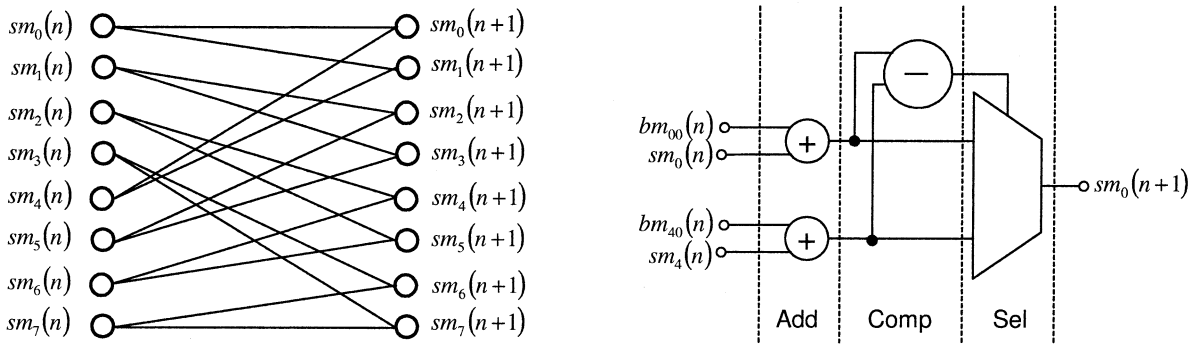


Fig. 4. Radix-2 trellis and ACS structure.

with large area penalties. The use of a redundant numbering system with MSB-first computations can provide performance improvement [10], [11]. However, this is achieved at the expense of large area due to the carry-save representation.

Previous high-throughput implementations of the Viterbi decoder [10]–[12] unrolled the ACS loop in order to perform two-step iterations of the trellis recursions within a single clock period. These lookahead methods replace the original radix-2 trellis (Fig. 4) with a radix-4 trellis (Fig. 5), at the cost of increased interconnect complexity. A radix-4 ACS computes four sums in parallel followed by a four-way comparison. In order to minimize the critical-path delay, the comparison is realized using six parallel pair-wise subtractions of the four output sums. In general, the critical-path delay increases. However, due to the doubled symbol rate, the effective throughput is improved if this increase in delay is less than twofold.

An alternative approach with a lower area overhead is the concurrent ACS [2]. Maintaining the use of a radix-2 trellis, the concurrent ACS performs the addition and comparison operations simultaneously. It requires the comparison to be realized with a four-input adder. A sub-8-ns four-input adder was implemented in 0.6-μm CMOS using two layers of three-to-two carry-save adders, followed by a final carry-lookahead adder. The critical path through the four-input adder and a multiplexer determines the throughput of the concurrent ACS.

Finally, an architecture, obtained through retiming and transformation of the ACS unit [13], [14] has a critical path consisting only of a two-input adder and a multiplexer. Without loss of generality, Fig. 6 shows a retimed ACS, which has been delayed by a third of a cycle. The operations are reordered as defined in (4). A comparison between the two sums is followed by selection of the appropriate minimum value, and finally, addition of the two corresponding branch metrics. The resulting structure has been labeled as a CSA unit. This transformation yields no performance gain: the subtraction no longer follows the addition and the carry profile is flattened by the multiplexer. The complete delays of the addition and subtraction appear in the critical path.

However, the CSA structure can be further transformed by moving the add operations before the select operation, as shown in Fig. 7, resulting in the parallel execution of the compare and add operations. The critical path delay is reduced to the combined delays of the comparator and the multiplexer. Although this modification incurs the cost of doubling the number of adders and multiplexers, it is less complex than the concurrent ACS.

$$sm_0(n+1) + bm_{00}(n+1) = \min \left\{ \begin{matrix} sm_0(n) + bm_{00}(n) \\ sm_4(n) + bm_{40}(n) \end{matrix} \right\} + bm_{00}(n+1). \quad (4)$$

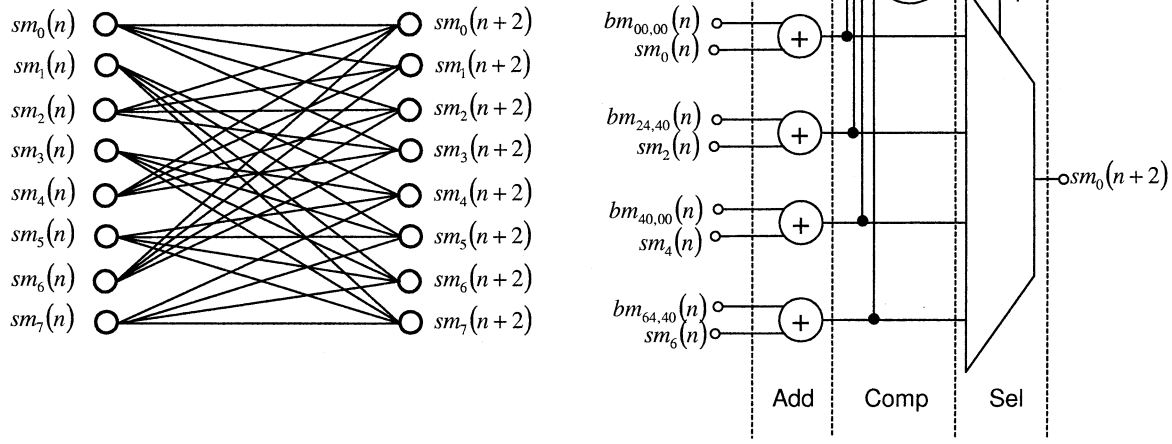


Fig. 5. Radix-4 trellis and ACS structure.

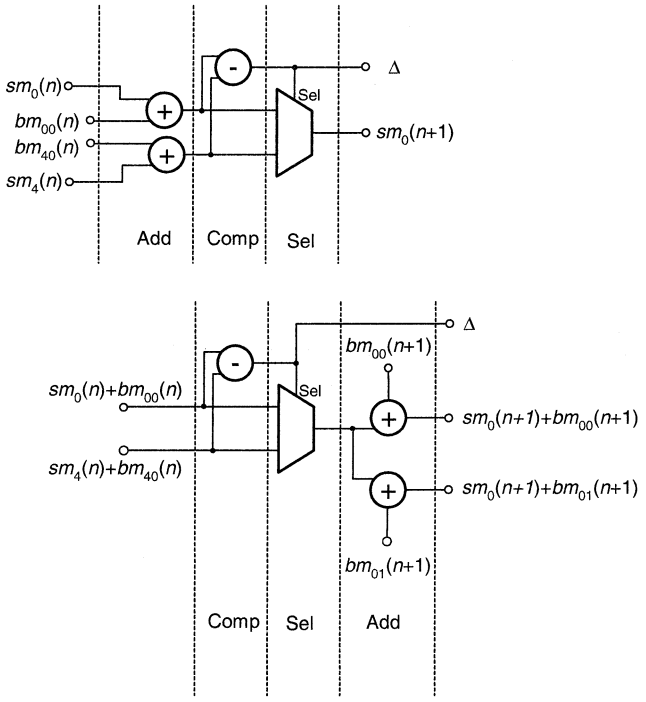


Fig. 6. Retiming of ACS units to construct CSA units.

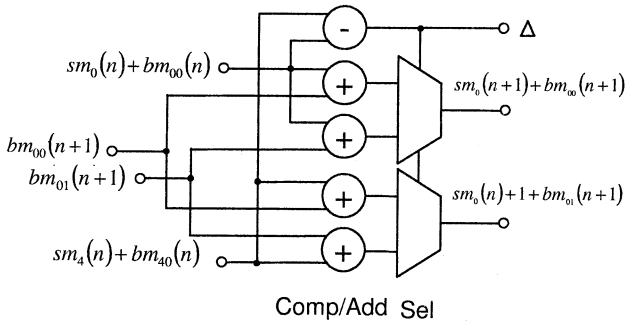


Fig. 7. Transformed compare-select-add (CSA) structure.

An exploration was performed to compare these ACS architectures using the radix-2 as the baseline. The various ACS structures were synthesized using low-threshold cells with high supply at best case conditions. The test was conducted on a block of eight ACSs, with interconnect resembling the underlying trellis structure. The decision outputs of the ACS structures were loaded with 200 fF to simulate the large capacitive load of the register exchange and FIFO memories. This work provides an analysis of the area-throughput and power-throughput tradeoffs across a range of permissible critical-path delay constraints. It differs from a prior comparison between the different ACS structures [15], which was mainly targeted toward minimum-delay implementations.

The power-throughput and area-throughput comparisons are plotted in Figs. 8 and 9. The synthesis algorithm [16] trades a higher area for delay reduction through sizing and logic transformations. Each curve tracks the same behavior. As the decreasing critical-path constraint approaches a minimum value, the area and power consumption of the synthesized structure increases sharply due to the use of increased gate sizes. The kinks in the curves correspond to points where logic transformations are preferred over increased sizing.

Table I shows a comparison of the relative throughput, power, and area of the test structures. The absolute numbers are dependent on the setup of the experiment such as the exact drive strengths of the inputs and capacitive output loads. However, the relative numbers are applicable for a wide range of operating conditions. As expected, the radix-4 ACS, which has been accounted for the doubled symbol rate, has the highest throughput. It is faster than the next fastest structure by a margin of 17%, but requires almost three times the area and two times the power. The radix-4 ACS is consistently larger and consumes more power than any of the other structures.

Both the transformed CSA and concurrent ACS are able to improve the throughput with significantly less area and power penalty. The choice of ACS structure is dependent on the required critical-path delay, and can be inferred from Figs. 8 and 9. The fanout-of-four (FO4) delay in this implementation technology is 50 ps. At this particular set of operating conditions,

TABLE I
MAXIMUM THROUGHPUT EFFICIENCY OF VARIOUS ACS ARCHITECTURES

	Relative Delay	Relative Symbol Throughput	Relative Area	Relative Power	Critical Path
Radix-2 ACS	1.00	1.00	1.00	1.00	(2 x 2-input Adders) + (1x Multiplexer)
Radix-2 Concurrent ACS	0.82	1.22	1.46	1.63	(1 x 4-input Adder) + (1 x Multiplexer)
Radix-2 CSA	0.71	1.41	1.99	1.89	(1 x 2-input Adder) + (1 x Multiplexer)
Radix-4 ACS	1.25	1.60	5.86	3.94	(1 x 2-input Adder) + (1 x 4-way comparator) + (1 x Multiplexer)

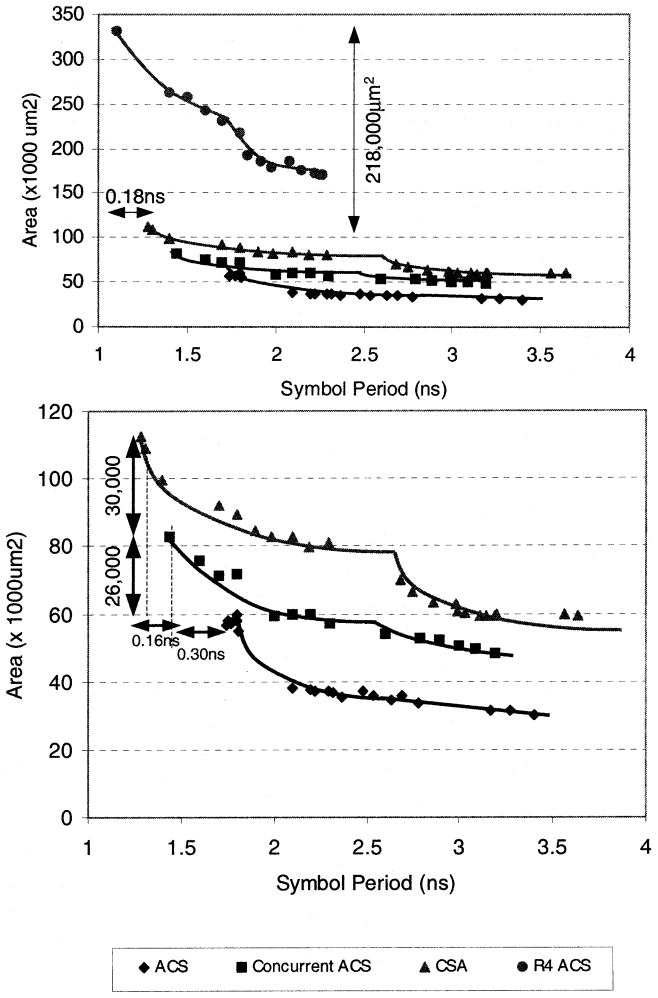


Fig. 8. Area comparisons of various ACS structures.

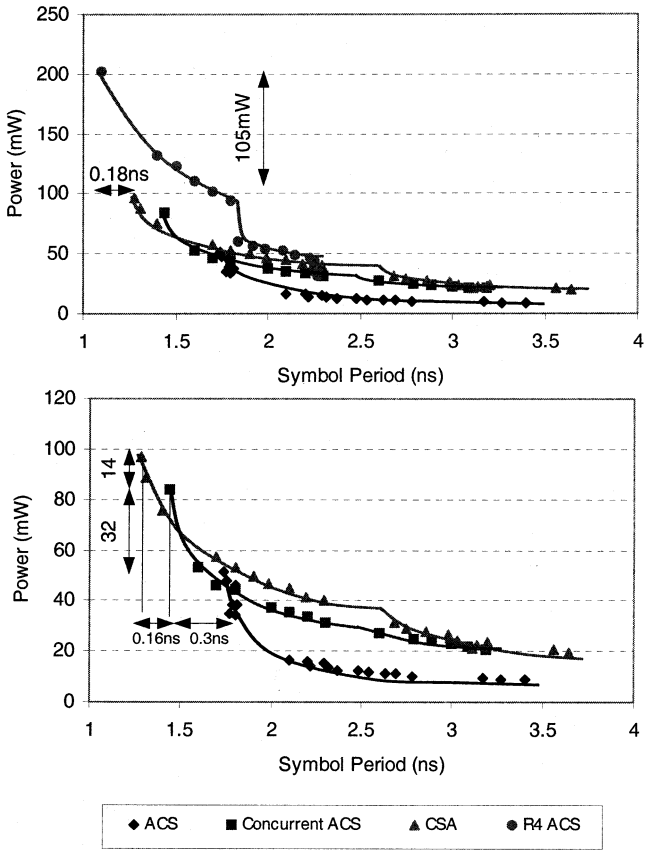


Fig. 9. Power comparisons of various ACS structures.

the transformed CSA structure is suitable for applications with critical-path delays specified between 26 and 29 FO4 delays. The concurrent ACS becomes the choice structure for delays between 29 and 35 FO4 delays. For low-throughput rates with critical-path delays above 35 FO4 delays, the ACS structure is the best choice in terms of both area and power consumption. In this high-throughput SOVA decoder implementation, the transformed CSA was implemented because it provided the highest decoding throughput, without incurring the excessive area and power penalties of the radix-4 ACS structure.

IV. SURVIVOR PATH DECODING

The two ML paths are determined by a two-stage traceback. An SMU is cascaded with a combination of a PED and an RMU. The SMU and PED have similar functions. Both essentially examine a list of competing paths by retracing a history of decisions and path metric differences. Previous implementations of the SOVA used either the register-exchange method [8] or memory-traceback method [15].

A. Register-Exchange and Memory-Traceback Methods

A register exchange consists of a two-dimensional array of one-bit registers and multiplexers as shown in Fig. 10. The registers in successive stages are interconnected to resemble the trellis structure of the convolutional code. A global clock signal controls the registers. The frequency of the clock determines the throughput of the Viterbi decoder. The path decision from each of the eight ACSs is input to the register-exchange pipeline and also selects the outputs of a corresponding row of multiplexers. At each clock cycle, a multiplexer located at row i and column k $\{i \in [1, 2, \ldots, 8], k \in [1, 2, \ldots, L]\}$ outputs a decision bit corresponding to a traceback of length k, originating from state i.

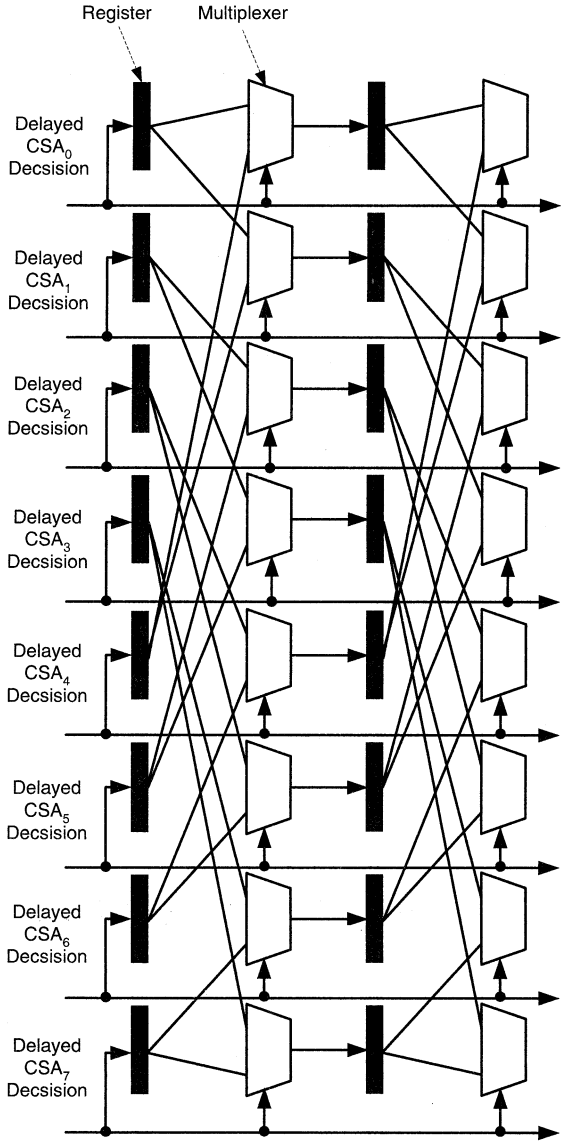


Fig. 10. Example eight-state register-exchange survivor memory unit used in VA-SMU.

This bit is stored in a register and will be input to a multiplexer at column $k+1$ in the following clock cycle.

The memory-traceback method has commonly been used in low-throughput low-power applications. It simply writes a vector of path decisions from the ACS recursions into RAM in each iteration of the Viterbi algorithm. After an initial startup delay, the decisions are retraced by reading the stored decisions in the reverse direction. Previous solutions have generally employed some variation of the k-pointer traceback architecture [15]. This technique uses $k-1$ parallel read pointers to access as many independent banks of memory, while a write pointer simultaneously stores the decisions from the ACS recursions into a k^{th} memory bank. An alternative [8] is to use a single bank of multiported DRAM.

The memory-traceback method permits the design of very compact RAM that provides significant area advantages. In 0.18-μm CMOS technology, the area of a typical SRAM cell is about 2.4 μm^2, in contrast with the 50-μm^2 area required for a flip-flop used in the register-exchange method.

The memory-traceback method stores the intermediate decision bits at static locations in memory. Since SRAM blocks typically operate by reading or writing multiple bits per cycle, a vector of decisions output by the parallel ACSs can be written into memory simultaneously. The traceback operation only needs to recall the decision bits that correspond to nodes along a particular traceback path. This contrasts with the register-exchange method, which constantly moves an array of decision bits through a pipeline of flip-flops.

In principle, this gives the memory-traceback method inherent power benefits. As the number of states rises, the register exchange is required to shift an increasing number of bits through its pipeline. However, for decoders with a small number of states, the use of standard SRAM modules offers little power or area advantage over register exchange because of the overhead of peripheral circuitry and standard word addressing [12]. Register exchange achieves high throughputs easily because its critical path only consists of a multiplexer and a register. Standard SRAM macros in 0.18-μm technology have much longer cycle times than the synthesized CSA recursion. Hence, the register exchange is the appropriate structure for high-throughput implementations of a decoder with a small number of states.

B. Path Equivalence Detector and Reliability Measure Unit

With the emphasis on high-throughput implementation, this section examines the use of register-exchange and the modifications necessary to implement the PED and RMU. The register-exchange method used in the SMU provides a convenient way to determine if competing traceback paths lead to equivalent decision bits. The two inputs to each multiplexer reflect the competing decisions, and a test for their equivalence can be realized by the addition of an XOR gate at each multiplexer location (Fig. 11). The ensuing Boolean outputs $\overline{EQ}_{ij}(n)$ indicate the equivalence between the two competing decisions obtained through a j-step traceback from state i.

From ACS_i, the difference between the two path metrics, $\Delta_i(n)$, arriving at time n, state i, is retained in FIFO buffers; $i \in \{1, 2, \ldots, 8\}$. 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 (Fig. 12).

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 binary representation of the reliability measure, 111111. Based on the $\overline{EQ}$ inputs, each pipelined section outputs one of the following:

$\overline{EQ} = 0$: Reliability measure from the previous step;
$\overline{EQ} = 1$: $\text{Min}\{\Delta_i$, Reliability measure from the previous step.$\}$

Compared with a Viterbi decoder implementation, the total size of the SMU and PED is approximately doubled ($L = M$). The RMU overhead includes M pipeline stages, each of which consists of a 2-input comparator with its Boolean output logically AND'd with the $\overline{EQ}_j$ input, a multiplexer, and a 6-bit register. The overall latency through the SOVA decoder is $L+M$. The additional latency remains insignificant compared to the

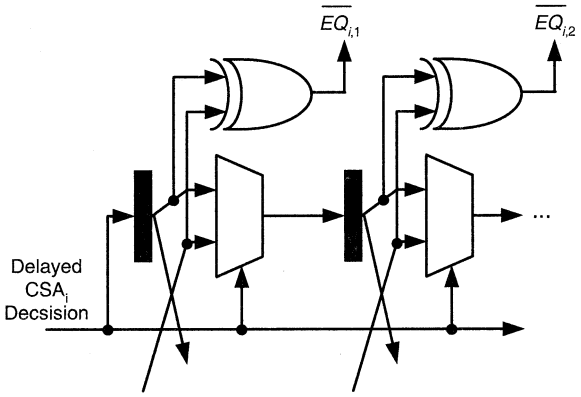


Fig. 11. State slice of register exchange used in the PED.

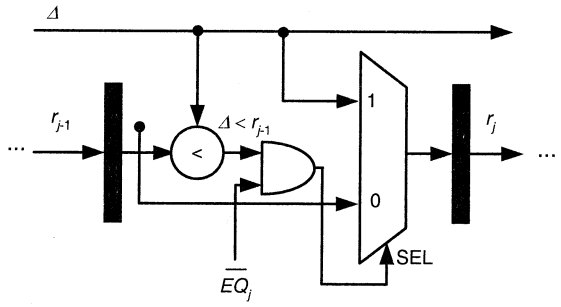


Fig. 12. Pipelined section of the RMU.

overall latency in the Turbo-SOVA system, which is dominated by the latency through the interleavers.

V. Design Flow and Test Procedures

The implementation of the SOVA decoders employed an automated design flow that performs direct mapping of signal processing algorithms into integrated circuits [18]. This automated flow was further enhanced for high-speed design through customization of the clock tree to achieve simulated clock skews that are less than 75 ps.

Both decoders have the same architecture, but are matched to different generator polynomials; the SOVA_EPR4 decoder is matched to an EPR4 channel with a $(1 \oplus D)^{-1}$ precoder while the SOVA_13 decoder is matched to an Octal(13) generator. The equivalent generator polynomials are $(1/(1 \oplus D))(1+D-D^2-D^3)$ and $(1 \oplus D^2 \oplus D^3)$, respectively.

The required wordlength of each SOVA decoder was ascertained by comparing the performance difference between floating-point computation and several fixed-point types. Seven-bit sign-magnitude signals were necessary to provide less than 0.1-dB degradation of required the SNR at a BER of 10^{-5} after five iterations.

The chip (Fig. 13) has been verified to decode data with 1.8-V supply at 25 °C. Throughput rates above 500 Mb/s were achieved and power dissipation was 400 mW. The speed characterization was performed using a clock tree with a built-in delay line. The speed and power performance of the EPR4 SOVA decoder is plotted in Fig. 14. The power measurements were performed at the highest frequencies permitted by the supply voltage. Table II summarizes the characteristics of the decoders. The above-average power dissipation can be attributed to the increased parallel CSA activities and the continuous movement of data through rows of shift registers and FIFO buffers. The latter is especially significant as the simulated clock power was 50% of the measured power consumed by the overall decoder.

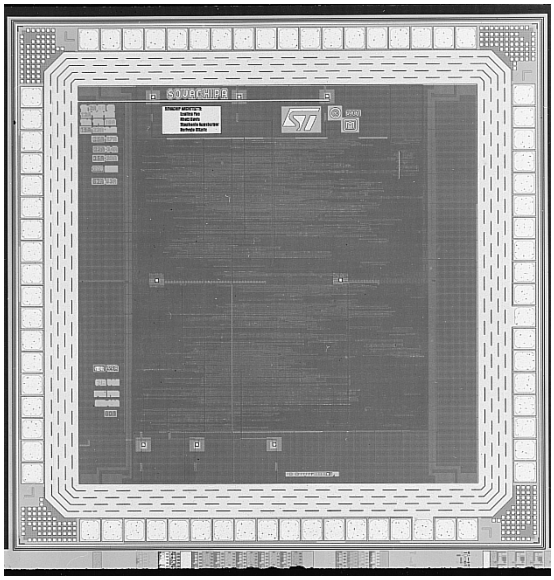


Fig. 13. Die micrograph.

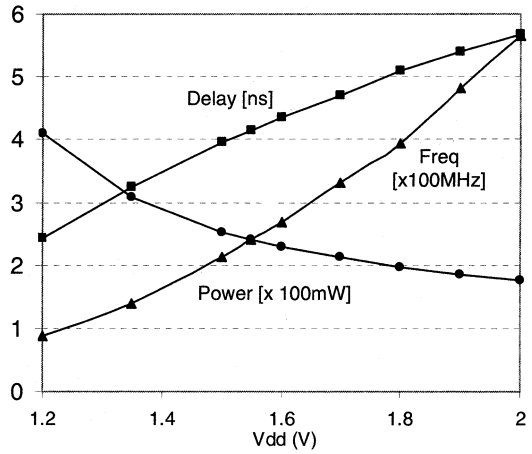


Fig. 14. Performance of EPR4 SOVA decoder.

TABLE II
Summary of Chip Implementation

Decoder Type	SOVA_EPR4	SOVA_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 @ 500MHz	395mW	400mW

VI. Conclusion

This design identifies the ACS recursion as the throughput bottleneck of the decoder design and compares four different structures for implementation of the ACS recursion. The results indicate that the preferred ACS structure varies as the crit-

ical-path delay constraint is relaxed. These inferences are applicable to the implementations of both soft and hard-decision Viterbi decoders. It was found that architectural retiming and transformation of the ACS structures with modification of the register exchange provided the highest throughput without excessive area and power penalties. Although the SOVA has less complexity than the MAP decoder, it still has higher power consumption than the hard-output Viterbi decoder. In practical high-performance iterative decoders, the power could be lowered through custom circuit design and technology scaling.

In addition to magnetic recording applications, the SOVA decoder can also be used in Turbo-coded forward error correction applications in high-throughput wireless, wireline, and optical communication systems.

ACKNOWLEDGMENT

The authors would like to thank T. Smilkstein, P. Pakzad, and V. Anantharam of UC Berkeley, and B. Gupta, S. Muroor, and B. Coates of ST Microelectronics for their technical assistance. The authors also thank ST Microelectronics for fabrication of the test chip.

REFERENCES

[1] T. V. Souvignier, M. Oberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for partial response channels," *IEEE Trans. Commun.*, vol. 48, pp. 1297–1308, Aug. 2000.

[2] S. Sridharan and L. R. Carley, "A 110 MHz 350 mW 0.6 mm CMOS 16-state generalized-target Viterbi detector for disk drive read channels," *IEEE J. Solid-State Circuits*, vol. 35, pp. 362–370, Mar. 2000.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[4] L. Bahl, J. Cocke, F. Jelinek, and R. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, pp. 284–287, Mar. 1974.

[5] J. Hagenauer and P. Hoecher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommunications Conf.*, Nov 1989, pp. 47.11–47.17.

[6] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Trans. Magnetics*, vol. 37, pp. 748–55, Mar. 2001.

[7] J. Hagenauer and L. Papke, "Decoding turbo-codes with the soft output Viterbi algorithm (SOVA)," in *Proc. IEEE Int. Symp. Information Theory*, July 1994, p. 164.

[8] O. J. Joeressen and H. Meyr, "A 40-Mb/s soft-output Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 30, pp. 812–818, July 1995.

[9] D. Garrett and M. Stan, "A 2.5-Mb/s, 23 mW SOVA traceback chip for turbo decoding applications," in *Proc. IEEE Int. Symp. Circuits and Systems* , May 2001, pp. 61–64.

[10] T. Gemmeke, M. Gansen, and T. Noll, "Implementation of scalable power and area efficient high-throughput Viterbi decoders," *IEEE J. Solid-State Circuits*, vol. 37, pp. 941–948, July 2002.

[11] A. K. Yeung and J. M. Rabaey, "A 210-Mb/s radix-4 bit-level pipelined Viterbi decoder," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 1995, pp. 88–89, 344, 440.

[12] P. Black and T. Meng, "A 1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE J. Solid-States Circuits*, vol. 32, pp. 797–805, June 1997.

[13] G. Fettweis, R. Karabed, P. H. Siegel, and H. K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signaling," in *Proc. IEEE Global Telecommunications Conf.*, Nov. 1995, pp. 559–563.

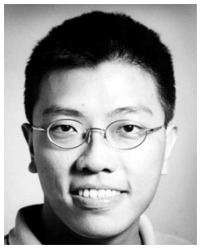
[14] I. Lee and J. L. Sonntag, "A new architecture for the fast Viterbi algorithm," in *Proc. IEEE Global Telecommunications Conf.*, Nov./Dec. 2000, pp. 1664–1668.

[15] T. Conway, "Implementation of high speed Viterbi detectors," *Electron. Lett.*, vol. 35, no. 24, pp. 2089–2090, Nov. 1999.

[16] *Design Compiler User Guide*, Synopsys Inc., Mountain View, CA, 2003.

[17] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Trans. Commun.*, vol. 41, pp. 425–429, Mar. 1993.

[18] W. R. Davis, N. Zhang, K. Camera, D. Markovic, T. Smilkstein, M. J. Ammer, E. Yeo, S. Augsburger, B. Nikolic, and R. W. Brodersen, "An automated design flow for high-throughput low-power dedicated signal processing systems," *IEEE J. Solid-State Circuits*, vol. 37, pp. 420–431, Mar. 2002.



Engling Yeo (S'96) received the B.S. and M.S. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1994 and 1995, respectively. He has been working toward the Ph.D degree in electrical engineering and computer science at the University of California, Berkeley, since 1999.

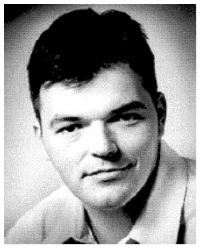
He was a Senior Member of Technical Staff in radar and signal processing systems at the DSO National Laboratories, Singapore, for three years. He also served as a Course Consultant for the National Technological University. His primary research interests are VLSI architectures for communication and storage systems.

Stephanie A. Augsburger (S'01) received the B.S. degree in computer engineering from Georgia Institute of Technology, Atlanta, in 2000 and the M.S. degree in electrical engineering from the University of California, Berkeley, in 2002, where she was funded by a Semiconductor Research Corporation Master's Scholarship.

She is currently a Member of Technical Staff at Intel Corporation, Santa Clara, CA, working on the design of Itanium processors. Her research interests are power optimization techniques for digital integrated circuits.

W. Rhett Davis (S'92–M'02) received the B.S. degrees in electrical and computer engineering from North Carolina State University, Raleigh, in 1994 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1997 and 2002, respectively.

He worked briefly with Hewlett-Packard (now Agilent), Boeblingen, Germany, and Chameleon Systems, San Jose, CA. Since 2002, he has been an Assistant Professor of electrical and computer engineering at North Carolina State University. His research interests are centered on improving methodologies and CAD tools for system-on-a-chip design. Other interests include low-power circuit design for communications and embedded systems.



Borivoje Nikolić (S'93–M'99) received the Dipl.Ing. and M.Sc. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1992 and 1994, respectively, and the Ph.D. degree from the University of California, Davis, in 1999.

He was on the faculty of the University of Belgrade from 1992 to 1996. He spent two years with Silicon Systems, Inc., Texas Instruments Storage Products Group, San Jose, CA, working on disk-drive signal processing electronics. In 1999, he joined the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, as an Assistant Professor. His research activities include high-speed and low-power digital integrated circuits and VLSI implementation of communications and signal-processing algorithms. He is a coauthor of *Digital Integrated Circuits: A Design Perspective* (2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 2003).

Dr. Nikolić received the National Science Foundation CAREER award in 2003, the College of Engineering Best Doctoral Dissertation Prize and the Anil K. Jain Prize for the Best Doctoral Dissertation in Electrical and Computer Engineering from the University of California, Davis, in 1999, and the City of Belgrade Award for the Best Diploma Thesis in 1992.