A 285-MHz Pipelined MAP Decoder in 0.18-μm CMOS

Seok-Jun Lee, *Member, IEEE*, Naresh R. Shanbhag, *Senior Member, IEEE*, and Andrew C. Singer, *Member, IEEE*

***Abstract*—Presented in this paper is a pipelined 285-MHz maximum *a posteriori* probability (MAP) decoder IC. The 8.7-mm^2 IC is implemented in a 1.8-V 0.18-μm CMOS technology and consumes 330 mW at maximum frequency. The MAP decoder chip features a block-interleaved pipelined architecture, which enables the pipelining of the add-compare-select kernels. Measured results indicate that a turbo decoder based on the presented MAP decoder core can achieve: 1) a decoding throughput of 27.6 Mb/s with an energy-efficiency of 2.36 nJ/b/iter; 2) the highest clock frequency compared to existing 0.18-μm designs with the smallest area; and 3) comparable throughput with an area reduction of 3-4.3$\times$ with reference to a look-ahead based high-speed design (Radix-4 design), and a parallel architecture.**

***Index Terms*—CMOS, iterative processing, maximum *a posteriori* probability (MAP) decoder, pipeline, turbo decoder, turbo equalizer.**

I. Introduction

SINCE its introduction in 1993 [1], turbo codes have found applications in several standards such as wideband CDMA (WCDMA), the 3rd Generation Partnership Project (3GPP) for IMT-2000 [2], Consultative Committee for Space Applications (CCSDS) telemetry channel coding [3], UMTS [4], DVB-RCS [5], IEEE 802.11n, and IEEE 802.16ab. The iterative techniques employed in turbo decoders have been extended to other receiver types such as turbo equalization [6] and turbo space-time decoders, all of which have shown remarkable (receiver) performance improvement over hard decision decoders and receivers with separate equalization and decoding [7]. We refer to receivers employing iterative techniques as *turbo receivers*.

Turbo receivers are composed of two or more constituent soft-input soft-output (SISO) decoders that communicate iteratively through an interleaver/de-interleaver. The SISO decoders employ the maximum *a posteriori* probability (MAP) algorithm as shown in Fig. 1 [7]. The throughput and complexity of turbo decoders and receivers is dominated by the SISO MAP decoders. Therefore, high-throughput implementations of MAP decoders are of great interest [8]–[18].

Extensive research has been done on investigating high-throughput implementations of turbo decoders and turbo equalizers [8]–[18]. These high-throughput architectures are based on parallel processing [8]–[16] and look-ahead computation [17], [18].

The parallel architectures partition a data block into M segments and process these segments in parallel [8]–[10]. Specialized data transfer and storage-management techniques and graph-theoretic approaches to model the resource time scheduling of MAP recursions have been employed to derive parallel high-throughput VLSI architectures [11]–[14]. The interleaver can also be structured in parallel architectures so that multiple memory accesses can be performed in parallel [15], [16], thereby further increasing the throughput. These approaches lead to an increase in hardware complexity that is linear with speedup.

Look-ahead architectures [17], [18] process multiple trellis sections per clock cycle. The state metric computations in the MAP algorithm are approximated to further decrease the critical path delay of the add-compare-select (ACS) recursions with an acceptable impact on the bit error rate (BER). However, look-ahead architectures have a complexity that increases exponentially with respect to throughput.

Each of these techniques achieves high throughput at the expense of silicon area. In this paper, we present the use of pipelining to enhance throughput with minimal impact on silicon area. We employ block-interleaved pipelining (BIP) [19], [20] at the architectural level to design a high-throughput MAP decoder chip in a 0.18-μm CMOS technology to achieve a 1.7$\times$–2.2$\times$ increase in clock frequency while simultaneously achieving a reduction in logic silicon area of 1.6 $\times$ –4.3$\times$ over existing 0.18-μm designs [12], [18], [21].

The remainder of this paper is organized as follows. Section II describes the pipelined MAP decoder architecture. The implementation of the MAP decoder IC is described in Section III, followed by a presentation of the results of testing the fabricated chip in Section IV.

II. Pipelined MAP Decoder Architecture

In this section, we first describe the BIP technique, followed by a brief description of the MAP algorithm and its data-flow graph, and then describe the architecture of the pipelined MAP decoder chip.

A. BIP Technique

Consider the recursive data path in Fig. 2(a). Note that the data path in Fig. 2(a) cannot be pipelined or parallelized due to the presence of the feedback loop. However, if the computations are block-based and the blocks can be partitioned into independent recursive subblock computations, then the block-

Manuscript received November 30, 2004; revised March 9, 2005. This work was supported by the National Science Foundation under Grant CCR 99-79381, Grant CCR-0092598, and Grant ITR 00-85929.

S.-J. Lee is with the Communication Systems Laboratory, DSP Solutions R&D Center, Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: s-lee8@ti.com).

N. R. Shanbhag and A. C. Singer are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: shanbhag@uiuc.edu; acsinger@uiuc.edu).

Digital Object Identifier 10.1109/JSSC.2005.852002

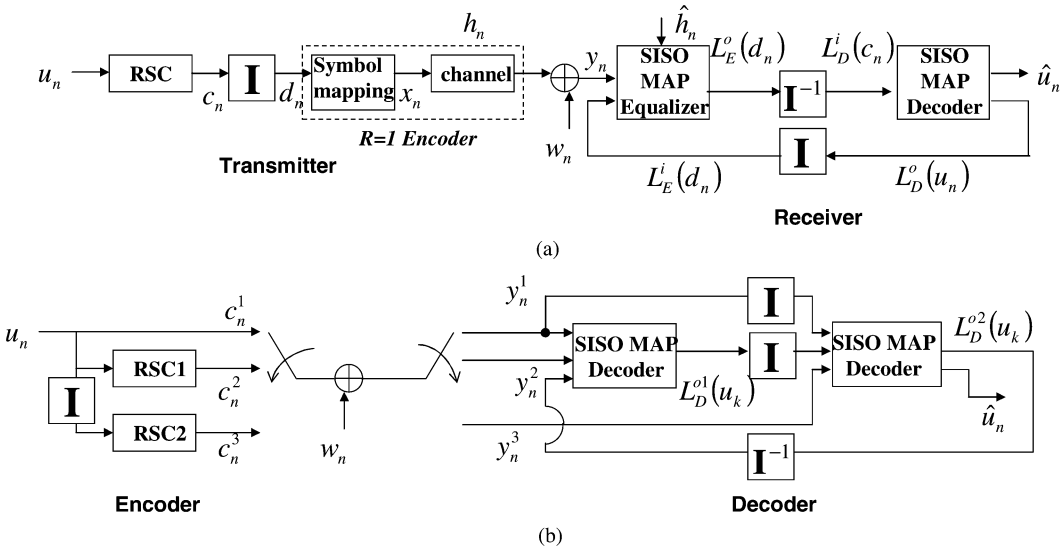


Fig. 1. Transmitter and receiver model in MAP-based turbo receivers, where $\mathbf{I}$ and $\mathbf{I}^{-1}$ denote block interleaving and de-interleaving, respectively, and a recursive systematic convolutional code (RSC) is employed. The subscripts E and D denote equalization and decoding, respectively, and the superscripts i and o refer to the input and output, respectively. (a) Turbo equalization. (b) Turbo decoding.

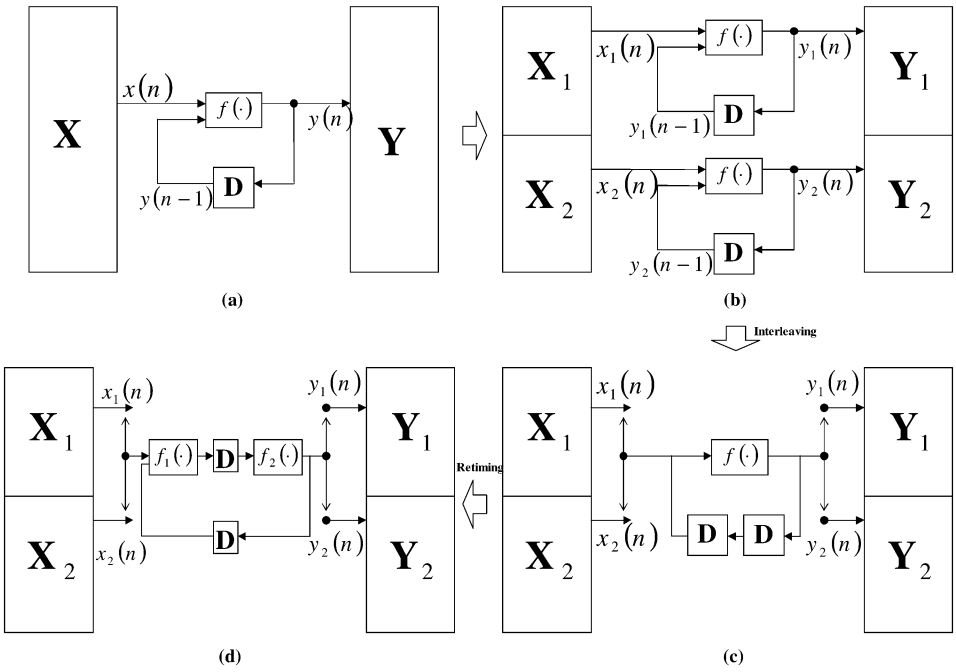


Fig. 2. Derivation of the block interleaved pipelined architecture. (a) Block-processing architecture. (b) Subblock parallel architecture. (c) Subblock interleaved pipelined architecture. (d) Block interleaved pipelined architecture.

processing architecture in Fig. 2(a) can be parallelized into an $M(M = 2)$ subblock parallel architecture [see Fig. 2(b)]. The subblock parallel architecture of Fig. 2(b) can be interleaved by a factor of M to obtain a subblock interleaved architecture shown in Fig. 2(c). The M *pipelining* latches thus introduced can be retimed as shown in Fig. 2(d) to generate the final block interleaved pipelined architecture. This process of exploiting the block processing nature of computations and the independence of subblock computations to pipeline an architecture is referred to as the BIP technique.

B. MAP Algorithm and Data-Flow Graph

A MAP decoder implements three types of computations: 1) computation of the branch metrics $\gamma_k(s', s)$; 2) computation of forward metrics a_k's and backward metrics b_k's; and 3) computation of the log-likelihood ratio $L(u_k)$ (LLR). These computations are described via the following equations:

$$\gamma_k(s', s) = \ln p(y_k|s, s') + \ln p(s|s') \tag{1}$$

$$a_k(s) = \max_{s'}{}^{*}[a_{k-1}(s') + \gamma_k(s', s)] \tag{2}$$

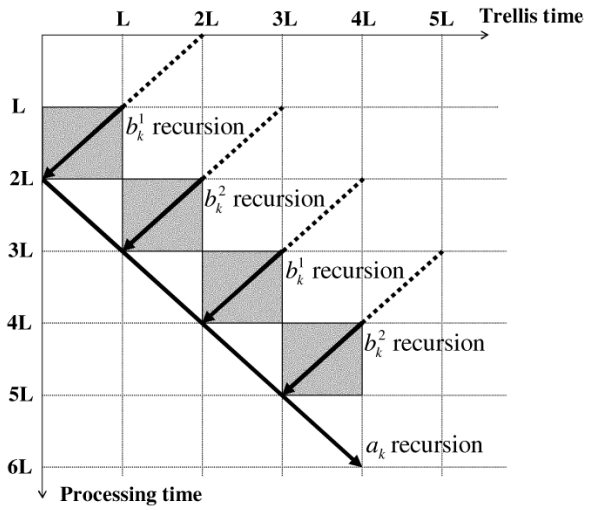


Fig. 3. Data-flow graph of the sliding-window log-MAP decoder. It is assumed that the computation and warm-up periods are equal to L.

$$b_{j-1}(s') = \overset{*}{\max_{s}}[b_j(s) + \gamma_j(s', s)] \tag{3}$$

$$L(u_k) \approx \overset{*}{\max_{s', s:u_k=1}}[a_{k-1}(s') + \gamma_k(s', s) + b_k(s)] - \overset{*}{\max_{s', s:u_k=0}}[a_{k-1}(s') + \gamma_k(s', s) + b_k(s)], \tag{4}$$

where k is the trellis index, u_k is the data at the kth trellis index, s and s' are trellis states, $L(u_k)$ is the LLR for u_k, and the max* operation is defined as

$$\overset{*}{\max}(x, y) \approx \max(x, y) + \ln(1 + e^{-|x-y|}). \tag{5}$$

Note that the computations described by (2) and (3) are nonlinear and recursive, which limits the achievable throughput. In this paper, we employ the BIP technique to pipeline the computations in (2) and (3), and the detailed derivation of BIP architecture for MAP decoding is well described in [19] and [20]. This BIP technique can be applied to the MAP decoder for any convolutional code regardless of its code polynomial.

We employ the sliding-window log-MAP algorithm [22] as it minimizes the metric storage requirements. Another advantage is that the log-MAP algorithm is formulated in terms of sums rather than in products, which leads to area saving in comparison with direct implementation of MAP algorithm. The sliding-window log-MAP algorithm can be derived via the *warm-up* property which states that the forward and backward metrics a_k and b_k converge after a few constraint lengths (K) have been traversed in the trellis, independent of the initial conditions. We refer to the warm-up period as L. The warm-up property can be employed only for computing backward metrics as shown in the data flow in Fig. 3, where the warm-up and computation period are depicted using dashed and solid lines, respectively.

Application of the warm-up property to the backward metric b_k computation results in the data flow shown in Fig. 4 where the block size $N = 10L$ and $M = 2$ subblocks of size $5L$ are considered. The L trellis sections between $4L$th and $5L$th nodes can be employed for generating reliable forward metric estimates for the second subblock [8]. No warm-up period is required for computing the metrics for the first and last trellis window as these are known to the decoder of terminated codes. For unterminated codes, the recursion can be initiated with equal state metrics. Note that the data flow in Fig. 4 satisfies the conditions necessary for the application of the BIP technique as described in Section II because the subblock computations are: 1) block-based and 2) recursive. Although the BIP technique is applied in a subblock level, we refer to this architecture as block-interleaved pipelined architecture where we assume that the block-size N and the subblock size N/M are multiples of the warm-up period L.

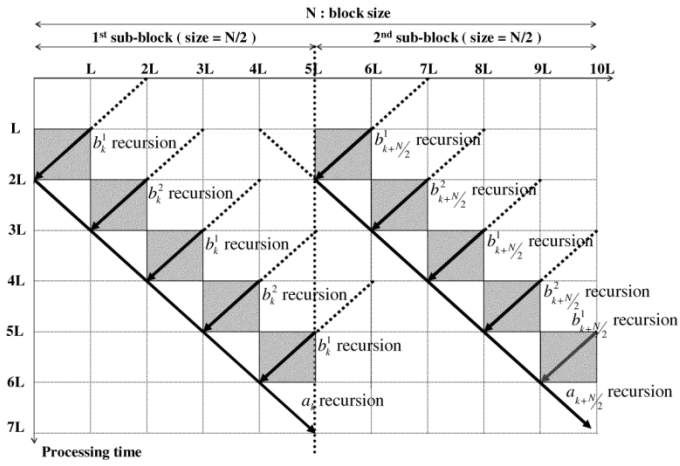


Fig. 4. Data-flow graph of subblock interleaved MAP computation.

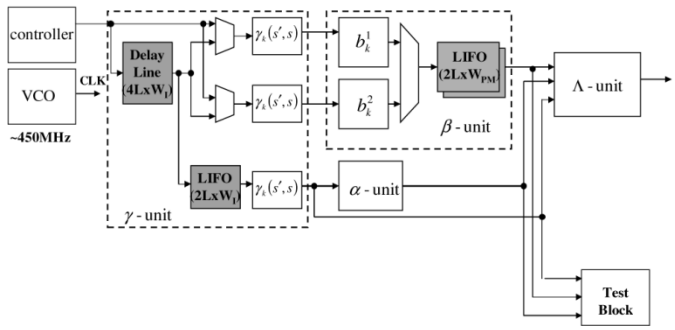


Fig. 5. Architecture of the MAP decoder chip, where W_I and $W_{\rm PM}$ denote the input data precision and the path-metric precision, respectively.

C. Architecture of the MAP Decoder Chip

The chip architecture of the MAP decoder is shown in Fig. 5. This architecture implements the log-MAP algorithm with a data-flow graph as shown in Fig. 4. This decoder architecture is composed of a γ-unit, an α-unit, a β-unit, and a Λ-unit. The γ-unit has one $4L \times W_I$ delay line, one $2L \times W_I$ last-input first-output buffer (LIFO), and three branch metric computation blocks, where W_I is the input data precision. The LIFO buffer is required to reorder the input sequence into the α-unit since the b_k^1 and b_k^2 units operate in a time reversed order. The α, b_k^1, and b_k^2 units have identical architectures as they implement a state-parallel update employing 2^{K-1} ACS kernels, where K is the constraint length as shown in Fig. 6(a). The β-unit also has 2^{K-1} LIFOs (each LIFO has the size of $2L \times W_{\rm PM}$) so that each LIFO stores intermediate $2L$ backward path metrics for each state. The Λ-unit computes the output LLR using γ_k, a_k,

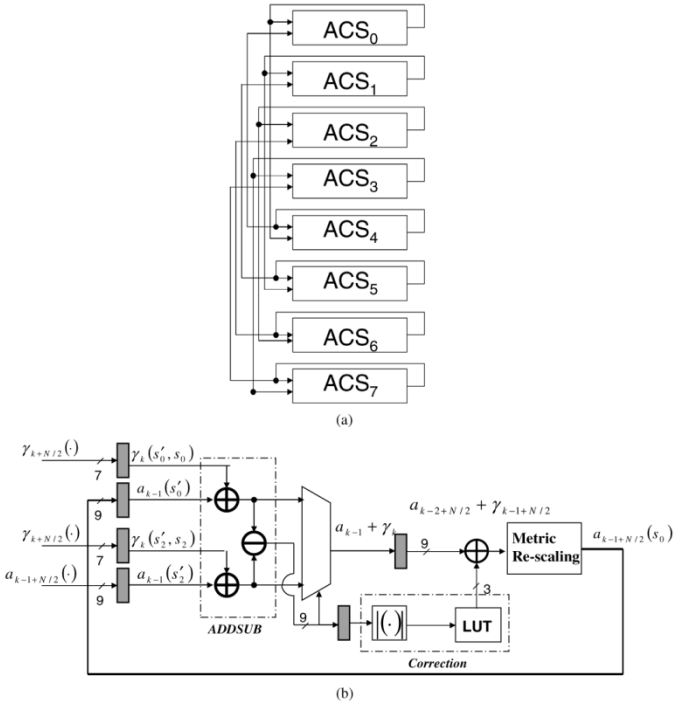


Fig. 6. State-parallel architecture of the α, b_k^1, and b_k^2 units: (a) block diagram and (b) the architecture of the pipelined ACS block.

and b_k as described in (4). The Λ-unit is not recursive and hence does not limit the throughput.

Note that the only recursive blocks in Fig. 5 are the α-unit and the b_k^1- and b_k^2-units. Assuming that the clock distribution and global interconnect are properly designed, it is the ACS units in the α-unit and the b_k^1- and b_k^2-units that will limit the maximum achievable throughput. In order to enhance the throughput, we apply the BIP technique described in Section II-A to pipeline the ACS architecture resulting in the pipelined ACS architecture shown in Fig. 6(b). The location of the pipelining latches are determined from circuit simulations. These latches are placed such that the critical path of the ACS datapath is split approximately in half.

D. Precision Requirements

The word lengths of internal variables of a MAP decoder are determined via simulations. The word lengths are chosen such that the loss in coding gain due to quantization effects is less than 0.05 dB at a BER of 10^{-4} when employed in a turbo decoder.

The precision of the path metrics in a MAP decoder should be determined carefully in order to avoid overflow. In this paper, we subtract a constant scale factor from all of the path metrics in order to avoid overflow [23].

The updated path metric at time index k is bounded by

$$\begin{aligned} a_k(s) &= \max_{s'}[a_{k-1}(s') + \gamma_k(s', s)] \\ &\leq Q + \lambda_{\max} \end{aligned} \tag{6}$$

where $a_{k-1} \leq Q$ and $\lambda_{\max}$ is the maximum branch metric. In practice, Q is not known but the upper bound $\Delta_{\rm pm}$ on the path metric difference between any two state metrics is known:

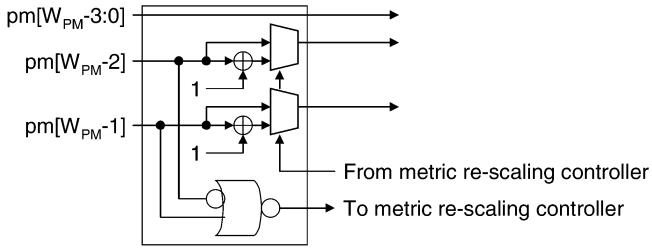


Fig. 7. Path metric rescaling logic, where the path metrics are represented in two's complement arithmetic.

$\Delta_{\rm pm} \leq \lambda_{\max} v$ [24], where v is the memory order of the encoder. Therefore, we choose the path metric precision, $W_{\rm PM}$, to satisfy $2^{(W_{\rm PM}-1)} > \Delta_{\rm pm} = \lambda_{\max} v$. This enables us to check for overflow of each path metric by using the circuit in Fig. 7.

If any path metric is greater than $2^{W_{\rm PM}-2}$, then we subtract $2^{W_{\rm PM}-2}$ from all state path metrics. In our design, the maximum branch metric $\lambda_{\max}$ is less than 2^6, v is 3, and $2^8 > \Delta_{\rm pm}$. Hence, $W_{\rm PM} = 9$ bits are assigned to the path metrics a_k and b_k. We also found empirically that 9 b of precision are sufficient to prevent overflow due to rescaling. The precision of all the signals in the decoder are shown in Fig. 6(b).

III. MAP Decoder IC Implementation

In this section, we first describe the design methodology, the floorplan and clock distribution network, and the design of the pipelined ACS datapath.

A. Design Methodology

We chose a full-custom design methodology for implementing the MAP decoder in order to fully exploit the benefits of using a 0.18-μm CMOS process. The MAP decoder was first implemented in VHDL and its functionality was verified using a commercial register transfer language (RTL) simulator.

Each subblock in the RTL behavioral description of the MAP architecture was translated into a transistor-level netlist using a schematic design tool. The corresponding layouts were drawn using a layout editor.

Layout-versus-schematic (LVS) and design-rule-checks (DRCs) were followed by timing verification using commercial static and dynamic timing analyzers at the subblock and full chip levels. Chip level verification included the input and output pads along with models of the off-chip load and drivers.

B. Floorplan and Clock Distribution Network

The chip floorplan shown in Fig. 8, where subblocks communicating to each other are placed close to each other, was designed to minimize the impact of interconnect delay on the critical path. Another factor contributing to the critical path delay are the setup and hold times of a latch. Data, control, and clock signals must be routed such that no setup and hold time violations occur. In order to prevent setup time violations, data and control signals are propagated in a direction opposite to that of the clock as shown in Fig. 8.

In order to minimize the hold-time, we employed C^2MOS latches, whose properties measured from circuit simulations are

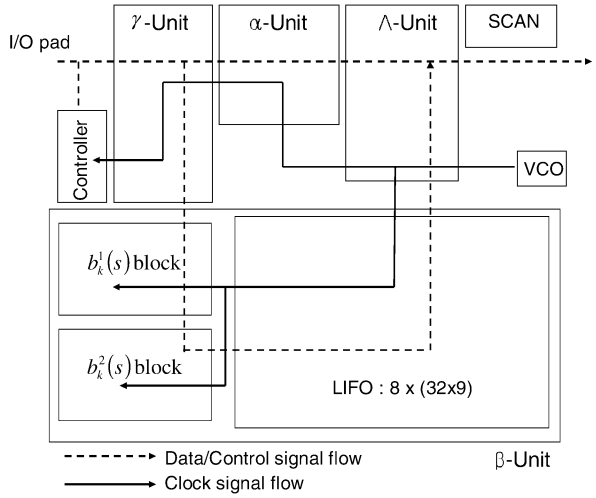


Fig. 8. Chip floorplan.

TABLE I
CHARACTERISTICS OF A C^2MOS LATCH WITH A 4 MINIMUM-SIZED INVERTER LOAD ($V_{DD} = 1.8$, TEMPERATURE = 110 °C, AND NOMINAL PROCESS CONDITION)

Setup time (ps)	80 (0→1), 100(1→0)
hold time (ps)	0(0→1), 0(1→0)
$CK - Q$ time (ps)	200((0→1), 100(1→0)

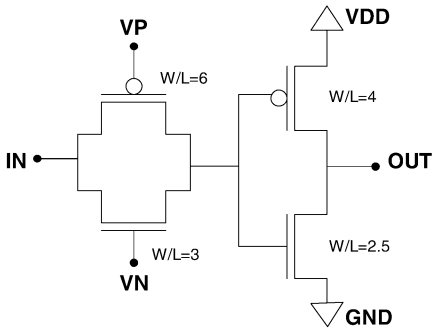


Fig. 9. Each stage circuit schematic of a ring oscillator in VCO.

shown in Table I. It should be noted that the C^2MOS latch has a hold time of zero. Thus, the combination of the opposite propagation of data and clock, minimization of clock skew, the use of zero hold time C^2MOS latches, and extensive timing verification over process corners provides confidence that no timing violations will occur.

In order to generate an internal clock, an 11-stage voltage-controlled oscillator (VCO) was implemented [25]. A VCO stage shown in Fig. 9 consists of a variable resistor at the input of an inverter. The resistor value is adjusted by controlling the gate voltages of the PMOS and NMOS transistors. The VCO tuning range is 1–540 MHz under various process corners as shown in Table II.

From timing analysis tools, we determined that the critical path occurs within the ACS kernel of the α and β units implying that inter subblock interconnect delay does not dominate. Thus, the ACS unit was designed carefully to minimize the critical path delay as described next.

TABLE II
MAXIMUM FREQUENCY OF AN 11-STAGE VCO ($V_{DD} = 1.8$, TEMPERATURE = 110 °C, AND NOMINAL PROCESS CONDITION)

Process Condition	Frequency (MHz)
SS	340
TT	430
FF	540

TABLE III
AREA AND SPEED OF THE ADDSUB BLOCK IN FIG. 6(b) WITH AN FO4 LOAD ($V_{DD} = 1.8$, TEMPERATURE = 110 °C, AND NOMINAL PROCESS CONDITION)

	RCA	CSA
Critical Path Delay (ns)	2.1	1.73
Area (μm^2)	7200	11448

C. ACS Datapath Design

The ACS datapath shown in Fig. 6(b) is mainly composed of additions. We chose a carry-select-adder (CSA) since it provides a good compromise between a fast high-complexity carry-lookahead adder and a slow low-complexity ripple-carry-adder (RCA). Table III shows the speed and area comparison between the CSA and RCA adders in 0.18-μm CMOS with a fan-out of four (FO4) load. The delay reduction is about 20% at a cost of 40% increase in area. The full adders in the ACS circuitry are implemented in the classic 24-transistor symmetric configuration.

The LUT is implemented using combinational logic with absolute value computation, where the logic gates are optimized to reduce delay and to save area. The metric rescaling logic was implemented as shown in Fig. 7 and a metric rescaling controller generates the enable signal whenever any of 2^{K-1} ACS path metrics is greater than 2^7.

Circuit simulations were conducted in order to compare the critical path delays of the pipelined and nonpipelined ACS data path. The critical path delay without pipelining is 4.56 ns and with pipelining is 2.68 ns.

IV. MEASURED RESULTS

The MAP decoder chip was designed and fabricated in 0.18-μm, 1.8-V CMOS process from TSMC with one poly and six metal layers and packaged in a 108-pin open cavity plastic package.

Low-frequency (<100 MHz) functional testing was done by using the external pattern generator and indicated that the chip functioned correctly with all the test vectors. High-frequency testing was done by using the on-chip VCO and pseudorandom binary sequence (PRBS) generator. The highest frequency at which all the test vectors pass was recorded as the operational frequency of the chip under the specified conditions. Via the test block, we probed intermediate values of each module and found

TABLE IV
MAP DECODER IC CHARACTERISTICS

Code polynomial	[13,15] in octal (K=3)
BIP pipeline depth	$M = 2$
Warm-up period	$L = 16$
Technology	TSMC 0.18μm 1P6M
Clock rate	285MHz @ 1.8V supply
Power consumption	330mW @ 285MHz
Transistor count	150K
Chip size	$3.467mm \times 2.513mm(8.7mm^2)$
Power supply	1.8V
Number of I/O Pins	72

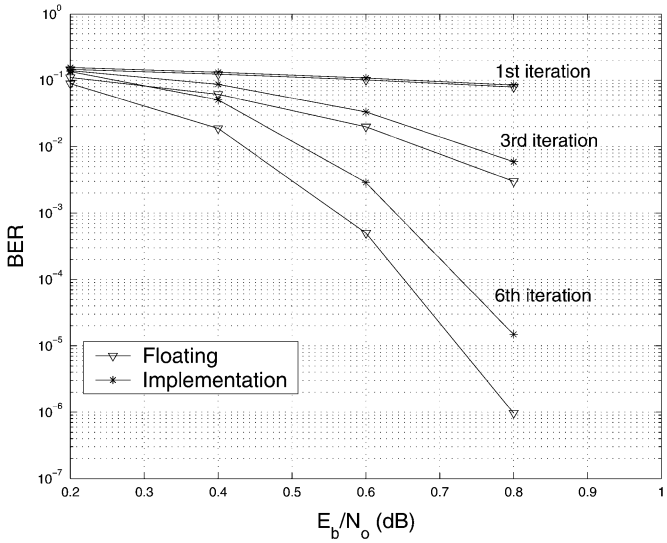


Fig. 10. BER performance when the designed MAP decoder ([13,15]$_8$) is embedded into Turbo decoder.

TABLE V
COMPARISON WITH THE STATE OF THE ART

Work	Process (μm)	f_{clk} (MHz)	Logic Area (core area) (mm^2)
This work	0.18	285	1.5(4.28)
[12]	0.18	170	4.55(7.16)
[18]	0.18	145	6.4(14.5)
[21]	0.18	128.8	2.42(9)

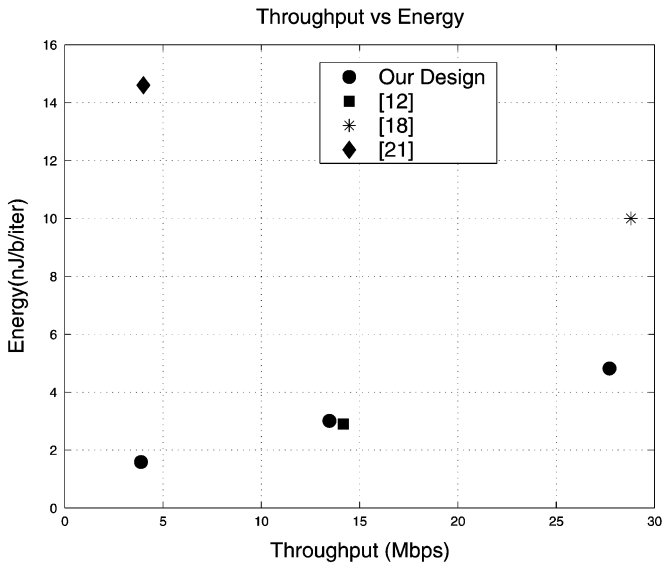


Fig. 11. Comparison of energy efficiency versus throughput. The energy consumption of our design includes the estimated power consumption of an interleaver (de-interleaver). The turbo decoder in [12] has 7 MAP kernels in parallel and 71.7-Mb/s maximum throughput. This figure includes the throughput number assuming two MAP kernels operate in parallel [12] since our design employs the $M = 2$ BIP architecture.

that b_k^1 or b_k^2 computation units of the β-unit failed first, and thus the critical path of the MAP decoder IC is in the ACS units of these blocks as planned during the design phase.

Table IV summarizes the features and measured characteristics the MAP decoder IC. The clock frequency of the MAP decoder is same as the clock frequency of a turbo decoder that employs the MAP decoder. The designed MAP decoder can be employed as MAP kernel for turbo decoder and equalizer applications. The clock frequency of 285 MHz is the fastest reported to date for a MAP or turbo decoder in 0.18-μm CMOS and represents an increase of $1.7\times$ –$2.2\times$ over existing implementations. Fig. 10 shows the BER performance when applied to turbo decoder applications.

Next, we compare the throughput and energy efficiency achieved by our MAP decoder IC in the context of a turbo decoder. A turbo decoder needs memory for interleaving and de-interleaving in addition to a MAP decoder core. As our chip does not include this memory, we estimated the memory power via using Texas Instruments 0.18-μm ASIC library, which indicates the number 0.018-μW/MHz/gate for typical design power dissipation. Combining the memory power estimate and the measured results indicate that a turbo decoder based on this MAP decoder core can achieve a decoding throughput of 27.6 Mb/s with an energy efficiency of 4.8175 nJ/b/iter assuming five iterations and 5 K interleaver size.

Table V shows the measured performance of the MAP decoder IC. The data rate and energy-efficiency numbers in Table V and Fig. 11 are those for a turbo decoder designed using the proposed MAP decoder core. Thus, as mentioned above, the energy-efficiency values for our work comprises the measured energy consumption of the MAP decoder and estimated energy consumption of interleaver/deinterleaver memory. Compared to the highest data-rate design [18], our decoder achieves comparable throughput with $4.3\times$ smaller logic area and a $2\times$ improvement in energy efficiency. Compared to the best low-power design [12], our decoder achieves a $1.9\times$ increase in throughput with comparable energy efficiency and a $3\times$ reduction in logic area. For the same throughput, an energy-efficiency improvement in the range of $1\times$–$7\times$ is achieved. Thus, the proposed MAP decoder core enables the design of a turbo decoder with the tradeoff between energy efficiency and throughput.

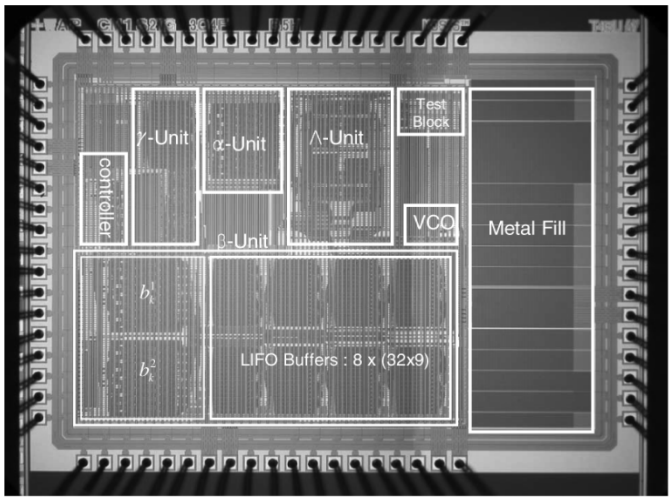


Fig. 12. Chip microphotograph.

The chip microphotograph is shown in Fig. 12.

V. Concluding Remarks

A pipelined MAP decoder IC was designed in TSMC 0.18-μm 1.8-V CMOS process technology. High-throughput operation was achieved via block-interleaved pipelining of the ACS kernel. The 8.7-mm^2 chip achieves a clock frequency of 285 MHz with 330 mW of power consumption at 1.8-V supply. This core achieves the highest clock frequency reported for any 0.18-μm designs and does so with the smallest area.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] Japan's Proposal for Candidate Radio Transmission Technology on IMT-2000: W-CDMA [Online]. Available: http://www.arib.or.jp/IMT-2000/proponent

[3] *Telemetry Channel Coding*, Consultative Committee for Space Data Systems (CCSDS), Blue book 101.0-B-4, May 1999.

[4] *Multiplexing and Channel Coding (FDD)*, Universal Mobile Telecommunications Systems (UMTS), 3G TS2 5.212 version 3.3.0 release 1999. [Online.] Available: http://www.etsi.org.

[5] C. Douillard, M. Jezequel, C. Berrou, N. Brengarth, J. Tousch, and N. Pham, "The turbo codec standard for DVB-RCS," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2000.

[6] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, 1995.

[7] J. Hagenauer, "Turbo principle: Tutorial introduction and state of the art," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Bretagne, France, 1997, pp. 1–11.

[8] Z. Wang, Z. Chi, and K. Parhi, "Area-efficient high-speed decoding schemes for turbo decoders," *IEEE Trans. Very Large Scale Integrat. (VLSI) Syst.*, vol. 10, no. 6, pp. 902–912, Dec. 2002.

[9] J. Hsu and C. Wang, "A parallel decoding scheme for turbo codes," in *Proc. IEEE Int. Conf. Circuits and Systems*, vol. 4, 1998, pp. 445–448.

[10] P. Urard *et al.*, "A generic 350 Mb/s turbo codec based on a 16-state turbo decoder," in *IEEE ISSCC Dig. Tech. Papers*, 2004, pp. 424–433.

[11] B. Bougard, A. Giulietti, L. V. d. Perre, and F. Catthoor, "A class of power efficient VLSI architectures for high speed turbo-decoding," in *Proc. IEEE Global Telecommunication Conf.*, vol. 1, 2002, pp. 549–553.

[12] B. Bougard *et al.*, "A scalable 8.7 nJ/bit 75.6 Mb/s parallel concatenated convolutional (turbo-) codec," in *IEEE ISSCC Dig. Tech. Papers*, 2003, pp. 152–153.

[13] A. Worm, H. Lamm, and N. Wehn, "Design of low-power high-speed maximum a priori decoder architectures," in *Proc. Design, Automation and Test in Europe, Conf. and Exhib.*, Apr. 2001, pp. 258–265.

[14] M. M. Mansour and N. R. Shanbhag, "VLSI architectures for SISO-APP decoders," *IEEE Trans. Very Large Scale Integrat. (VLSI) Syst.*, vol. 11, no. 4, pp. 627–650, Aug. 2003.

[15] A. Giulietti *et al.*, "Parallel turbo code interleavers: Avoiding collisions in accesses to storage elements," *Electron. Lett.*, vol. 38, no. 5, pp. 232–234, Feb. 2002.

[16] B. Bougard, A. Giulietti, C. Desset, L. Van der Perre, and F. Catthoor, "A low power high speed parallel concatenated turbo-decoding architecture," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2003, pp. 511–514.

[17] T. Miyauchi, K. Yamamoto, and T. Yokokawa, "High-performance programmable SISO decoder VLSI implementation for decoding turbo codes," in *Proc. IEEE Global Telecommunications Conf.*, vol. 1, 2001, pp. 305–309.

[18] M. Bickerstaff, L. Davis, C. Thomas, D. Garret, and C. Nicol, "A 24 Mb/s radix-4 LogMAP turbo decoder for 3 GPP-HSDPA mobile wireless," in *IEEE ISSCC Dig. Tech. Papers*, 2003, pp. 150–151.

[19] S. Lee, N. Shanbhag, and A. Singer, "A low-power VLSI architecture for turbo decoding," in *Proc. Int. Symp. Low Power Electronics and Design*, Aug. 2003, pp. 366–371.

[20] ——, "Area-efficient, high-throughput MAP decoder architectures," *IEEE Trans. Very Large Scale Integrat. (VLSI) Syst.*, to be published.

[21] M. Bickerstaff, D. Garrett, T. Prokop, C. Thomas, B. Widdup, G. Zhou, L. Davis, G. Woodward, C. Nicol, and R. Yan, "A unified turbo/Viterbi channel decoder for 3 GPP mobile wireless in 0.18-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1555–1564, Nov. 2002.

[22] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 260–264, Feb. 1998.

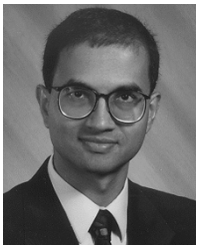
[23] C. B. Shung, P. H. Siegel, G. Ungerboeck, and H. K. Thapar, "VLSI architecture for metric normalization in the Viterbi algorithm," in *Proc. IEEE Int. Conf. Communications*, vol. 4, 1990, pp. 1723–1728.

[24] A. P. Hekstra, "On the maximum difference between path metrics in a Viterbi decoder," in *Proc. IEEE Int. Symp. Information Theory*, 1993, pp. 21–21.

[25] N. Retdian, S. Takagi, and N. Fujii, "Voltage controlled ring oscillator with wide tuning range and fast voltage swing," in *Proc. IEEE Asia-Pacific Conf. ASICs*, Aug. 2002, pp. 201–204.

Seok-Jun Lee (S'96–M'98) was born in Pusan, Korea, in 1973. He received the B.S. and M.S. degrees from Seoul National University (SNU), Seoul, Korea, in 1996 and 1998, respectively, both in electrical engineering, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2004.

During the summers of 2001 and 2002, he was with the Communication Systems Laboratory, DSPS R&D Center, Texas Instruments, Inc., Dallas, TX, where he was involved in IEEE 802.11 and 802.16 modem development. Since September 2004, he has been working on DVB-H and ISDB-T modem development. His research interests include VLSI architectures for communication and signal processing systems.



Naresh R. Shanbhag (M'88–SM'98) received the B.Tech. degree from the Indian Institute of Technology, New Delhi, India, in 1988, the M.S. degree from the Wright State University, Dayton, OH, in 1990, and the Ph.D. degree from the University of Minnesota, Minneapolis, 1993, all in electrical engineering.

From 1993 to 1995, he was with AT&T Bell Laboratories, Murray Hill, NJ, where he was the Lead Chip Architect for AT&T's 51.84 Mb/s transceiver chips over twisted-pair wiring for Asynchronous Transfer Mode (ATM)-LAN and very high-speed digital subscriber line (VDSL) chip-sets. Since August 1995, he has been with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, where he is presently a Professor. His research interests are in the design of integrated circuits and systems for broadband communications, including low-power/high-performance VLSI architectures for error-control coding, equalization, as well as digital integrated circuit design. He has published more than 90 journal papers/book chapters/conference publications in this area and holds three U.S. patents. He is also a coauthor of the research monograph *Pipelined Adaptive Digital Filters* (Norwell, MA: Kluwer, 1994).

Dr. Shanbhag was the recipient of the 2001 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Best Paper Award, the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the 1999 Xerox Faculty Award, the National Science Foundation CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems Society. Since July 1997, he has been a Distinguished Lecturer for the IEEE Circuits and Systems Society. From 1997 to 1999 and from 1999 to 2002, he served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, respectively. He has served on the technical program committees of various conferences.

Andrew C. Singer (M'98) was born in Akron, OH, in 1967. He received the S.B., S.M., and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 1990, 1992, and 1996, respectively, all in electrical engineering and computer science.

Since 1998, he has been on the faculty of the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, where he is currently an Associate Professor in the Electrical and Computer Engineering Department and a Research Associate Professor in the Coordinated Science Laboratory. During the academic year 1996, he was a Postdoctoral Research Affiliate with the Research Laboratory of Electronics at MIT. From 1996 to 1998, he was a Research Scientist at Sanders, A Lockheed Martin Company, Manchester, NH. His research interests include statistical signal processing and communication, universal prediction and data compression, and machine learning.

Prof. Singer was a Hughes Aircraft Masters Fellow and was the recipient of the Harold L. Hazen Memorial Award for excellence in teaching in 1991. In 2000, he was the recipient of the National Science Foundation CAREER Award and in 2001 the Xerox Faculty Research Award. He is currently a member of the MIT Educational Council, Eta Kappa Nu, and Tau Beta Pi.