A Low Power Turbo/Viterbi Decoder for 3GPP2 Applications

Chien-Ching Lin, Yen-Hsu Shih, Hsie-Chia Chang, and Chen-Yi Lee

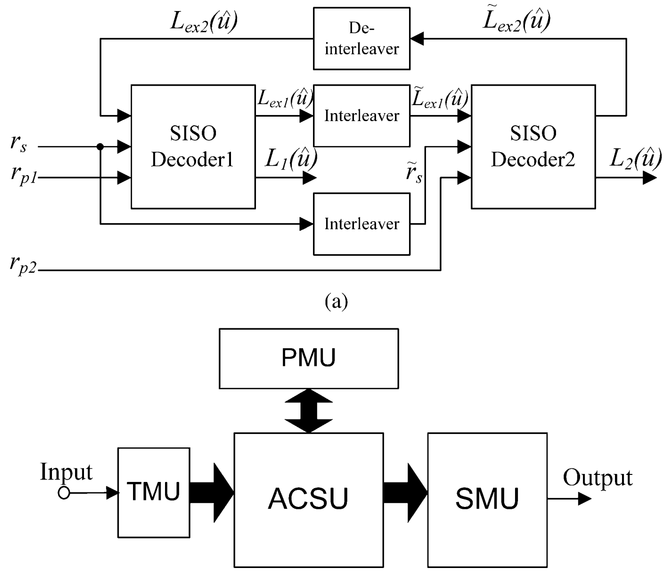
***Abstract*—This paper presents a channel decoder that completes both turbo and Viterbi decodings, which are pervasive in many wireless communication systems, especially those that require very low signal-to-noise ratios. The trellis decoding algorithm merges them with less redundancy. However, the implementation is still challenging due to the power consumption in wearable devices. This research investigates an optimized memory scheme and rescheduled data flow to reduce power consumption and chip area. The memory access is reduced by buffering the input symbols, and the area is reduced by reducing the embedded interleaver memory. A test chip is fabricated in a 1.8 V 0.18-μm standard CMOS technology and verified to provide 4.25-Mb/s turbo decoding and 5.26-Mb/s Viterbi decoding. The measured power dissipation is 83 mW, while decoding a 3.1 Mb/s turbo encoded data stream with six iterations for each block. The power consumption in Viterbi decoding is 25.1 mW in the 1-Mb/s data rate. The measurement shows the power dissipation is 83 mW for the turbo decoding with six iterations at 3.1 Mb/s, and 25.1 mW for the Viterbi decoding at 1 Mb/s.**

***Index Terms*—Cache memories, error correction, mobile communication, turbo, Viterbi decoding.**

I. Introduction

The parallel concatenated convolutional codes (PCCC), named turbo code [1], has been widely adopted in wireless communication systems. Turbo code can achieve an excellent coding performance with simple constituent codes concatenated by an interleaver whose length N will reduce the bit error rate (BER) by a factor of $1/N$ [2]. In addition, convolutional codes are simple and practical error correcting codes, and the Viterbi algorithm is an optimal solution for decoding them. The coding gain of turbo codes is better than that of convolutional codes on the basis of comparable complexity. However, the iterative decoding in turbo decoders limits the decoding speed and increases the decoding latency.

Fig. 1 illustrates the decoding flow of turbo and Viterbi decoders. Turbo decoder (TD) consists of two soft-in/soft-out (SISO) decoders based on either the soft-output Viterbi algorithm (SOVA) [3] or the maximum *a posteriori* probability (MAP) algorithm. The MAP algorithm, also referred to as the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [4], is the optimal symbol-by-symbol detection algorithm that minimizes the error probability, but is much more complex than the SOVA. Hence, two approximations, Log-MAP and Max-Log-MAP [5] algorithms, were proposed and proved to be more practical in circuit implementation. In Fig. 1(a), the decoding proceeds iteratively with the extrinsic information $L_{\rm ex1}(\hat{u})$ and $L_{\rm ex2}(\hat{u})$ passing between two SISO decoders separated by the interleaver. Finally, the log-likelihood ratio (LLR) $L(\hat{u})$ is exported for hard decisions. In Fig. 1(b), the Viterbi decoder (VD) contains four main units: transition metric unit (TMU), add-compare-select unit (ACSU), path metric unit (PMU), and survivor memory unit (SMU). TMU calculates the transition metric (TM) from the input data. ACSU recursively accumulates TMs as path metrics (PMs), and makes decisions to select the most likely state transition. Finally, SMU traces the decisions to extract decoded data. In third-generation (3G) mobile wireless communication [6], both turbo and convolutional codes are specified for high-speed data and speech transmission. Higher data rates and larger block lengths in the turbo code indicate more design challenges due to large memory size and bandwidth.



(a)

(b)

Fig. 1. Decoding flowchart. (a) Turbo decoding. (b) Viterbi decoding.

In [7] and [8], the memory blocks are optimized to achieve a significant power reduction. The suboptimal approaches that reduce the number of states or paths in trellis are also presented as power saving techniques, but the performance becomes degraded. The turbo decoders with the block length 5114 are also reported in [9] and [10]. For 3G application, the integration of turbo and Viterbi decoders is also reported in [9]. However, there is little research available on the implementation of the large turbo code in 3GPP2 system [6]. We present a channel decoder that integrates both turbo and Viterbi decodings with the optimized memory organization, as well as the low-power dissipation. The turbo decoder is designed with a single SISO decoder architecture based on the Max-Log-MAP algorithm, and the embedded interleaver is implemented with the modest memory size. It also features a cache buffer to increase the bandwidth efficiency for the SISO decoder and reduce the external memory access. The Viterbi decoder uses the SISO decoder to perform 256 ACS computations. The interleaver memory in turbo decoder is shared with the SMU, and an additional PMU is designed to store 256 PMs. The TMU also supports four coding rates defined in [6].

This paper is organized as follows. The decoding algorithm is described in Section II. Section III presents the decoder architecture. The chip implementation and test results are shown in Section IV. Finally, a conclusion is given in Section V.

II. Algorithm

A. Turbo Decoding Algorithm

The algorithm iteratively decodes the PCCC with the MAP algorithm that calculates *a posteriori* probability (APP) of each information bit u_k [4]. In logarithmic domain, the modified BCJR [1] algorithm is applied to generate the log-likelihood radio (LLR) of APP $L(\hat{u}_k)$. For systematic codes, $L(\hat{u}_k)$ can be further expressed in three terms [11]

$$L(\hat{u}_k) = L_{\rm in}(u_k) + L_c r_s + L_{\rm ex}(\hat{u}_k). \qquad (1)$$

These are channel value $L_c r_s$, *a priori* information $L_{\rm in}(u_k)$, and extrinsic information $L_{\rm ex}(\hat{u}_k)$. $L_{\rm in}(u_k)$ comes from $L_{\rm ex}(\hat{u}_k)$ of the other SISO decoder after interleaving or de-interleaving.

Manuscript received January 29, 2005; revised November 13, 2005 and January 9, 2006. This work was supported in part by the National Science Council under Grant 94-2215-E-007046 and by the Ministry of Economic Affairs, Taiwan, under Grant 93-EC-17-A-03-S1-005.

The authors are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C. (e-mail: cclin@si2lab.org).

Digital Object Identifier 10.1109/TVLSI.2006.874375

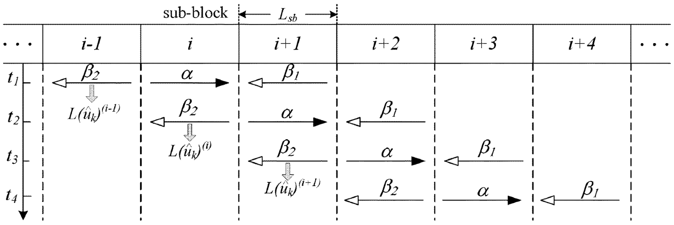


Fig. 2. Windowed MAP algorithm.

To reduce the memory requirement in SISO decoders, the sliding window algorithm is applied [12]. In Fig. 2, the data stream is divided into sub-blocks of length $L_{\rm sb}$, and the dummy backward recursion β_1 is employed to find the beginning of states for the true backward recursion β_2. The final results are computed from α, β_2, and γ, and the extrinsic information is also generated to the other SISO decoder.

B. Viterbi Algorithm

The algorithm recursively computes PMs for the shortest path terminating at each state. With a truncation length of T, the algorithm must decide a state at depth $t-T$ by the shortest path at depth t. Besides, the traceback approach based on the k-point even algorithm [13] is applied for the low-power consideration.

III. Architecture Design

The trellis decoding structure of both decoder enables the resource sharing of the ACS and the memory units, leading to the area efficient architecture.

A. Turbo Decoder

Fig. 3 shows the decoder architecture in the turbo mode where the active components are highlighted. The TD is a single SISO decoder architecture consisting of three ACS groups for α, β_1 and β_2 recursions in Fig. 2, and each ACS group contains eight ACS units. The SISO decoder processes three consecutive sub-blocks concurrently for different strategies in the windowed MAP algorithm. ACS-α carries out the forward recursion and saves the results in SRAM-α. ACS-β_2 starts backward recursion from the state previously determined by ACS-β_1. At the same time, LLR unit calculates the $L(\hat{u}_k)$ and $L_{ex}(\hat{u}_k)$ which is formulated in (1).

B. Cache Memory Design

In Fig. 2, the data of a sub-block needs to be read by ACS-β_1, ACS-α, and ACS-β_2 units separately. At the same time slot, from t_i to t_{i+1}, three consecutive sub-blocks are read by the TMUs. The minimum data bandwidth to the external codeword memory should be $3f_c$M symbols per second (MS/s), assuming a f_c MHz working frequency in the ACS units. Thus, an input cache is implemented to reduce the repeated access of the external memory. With four banks memory model, the behavior of each bank can be expressed by Fig. 4, where each bank has $L_{\rm sb}$ words and should be connected to the three TMUs with multiplexers. Codewords are written to the memory and read by the TMU-α, the TMU-β_1, and the TMU-β_2 for TM calculations. The data bandwidth of the cache is f_c MS/s for inputs and $3f_c$ MS/s for outputs. Accordingly, a multiport memory or a higher working frequency can be applied to reduce the interconnection between the cache and the TMUs. However, both methods may lead to larger area or more power consumption. Hence, as shown in Fig. 5, we use a hybrid cache solution where a dual-port memory works at the double clock frequency to provide the quadruple-port function in Fig. 4(a). The reading by TMU-β_2 and the codeword writing is further combined by avoiding the write-after-read (WAR) data hazard; as a result, the memory size can be reduced from $4L_{\rm sb}$ to $3L_{\rm sb}$ words.

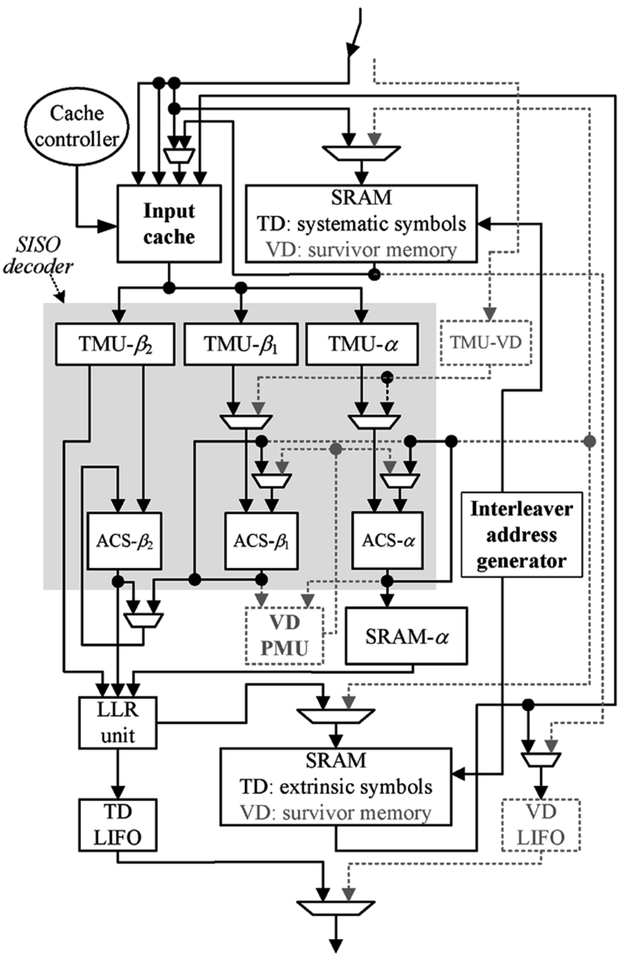


Fig. 3. Decoder in the turbo mode.

C. Interleaver Design

The embedded interleaver/de-interleaver is designed to reduce the amount of time required to permute symbols. The larger block length N can achieve lower bit error rate (BER), but requires more memory, as well as chip area. In this design, a single memory block is used for both interleaving and de-interleaving functions. In SISO decoder1, the extrinsic information is read and written in a sequential order, while the extrinsic information is accessed in a permuted order in SISO decoder2. Therefore, the data in the memory are always in sequence regardless of the permutation. Note that the SISO decoder in Fig. 3 performs both SISO decoder1 and SISO decoder2 functions in different time slots, leading to no data hazard. The memory requires one reading port and one writing port in this configuration, and can be either a dual-port SRAM (DP-SRAM) or a single-port SRAM (SP-SRAM) working at higher clock rates.

The permutation realized by address management operates on-the-fly with the SISO decoder and induces no additional delay within each iteration. However, in some cases [6], the address generator (AG) may produce invalid addresses and stall the SISO decoder. This can be solved by using two AGs, as illustrated in Fig. 6. While an invalid address is observed, the address from the other generator is adopted.

D. Viterbi Decoder

In the Viterbi mode, 256 states trellis decoding is implemented with $1/2, 1/3, 1/4$, and $1/6$ coding rates. As shown in Fig. 7, the ACS-α and ACS-β_1 that contain 16 ACS units, perform these 256 ACS operations in 16 cycles. The memory for the interleaver of TD is treated as the survivor memory. The traceback (TB) read operation is performed

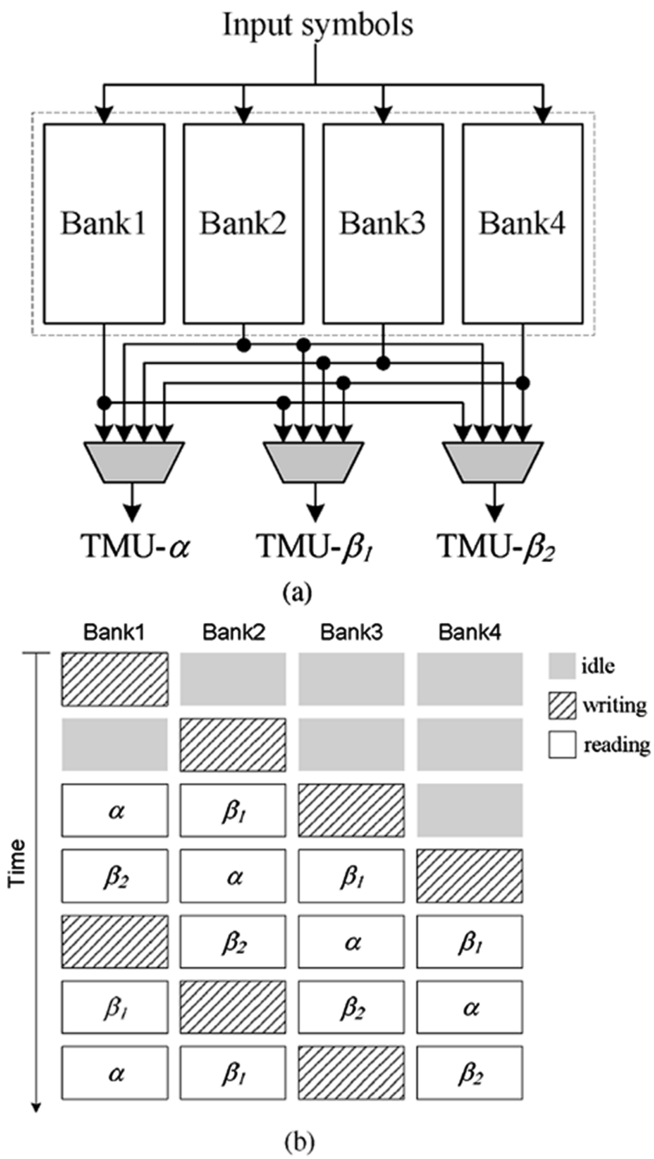


Fig. 4. Multibank cache model.

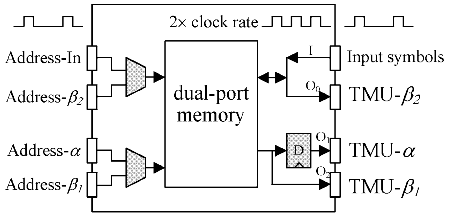


Fig. 5. Quadruple-port cache memory architecture.

separately from ACS operations due to the limited memory bandwidth and takes an additional two cycles based on the 3-point even algorithm [13], [14]. The decode read follows the second traceback read and outputs a decoded bit.

The decoding flow is illustrated in Fig. 8. On average, to decode one data bit, it takes 19 cycles where the ACS units take 16 cycles to write new decisions, the TB read operation spends two cycles, and the decode read operation needs one cycle. In a 100-MHz clock rate, the Viterbi decoder can achieve the maximum throughput of 5.26 Mb/s.

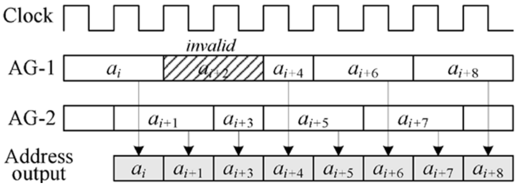


Fig. 6. Removal of invalid addresses with two AGs.

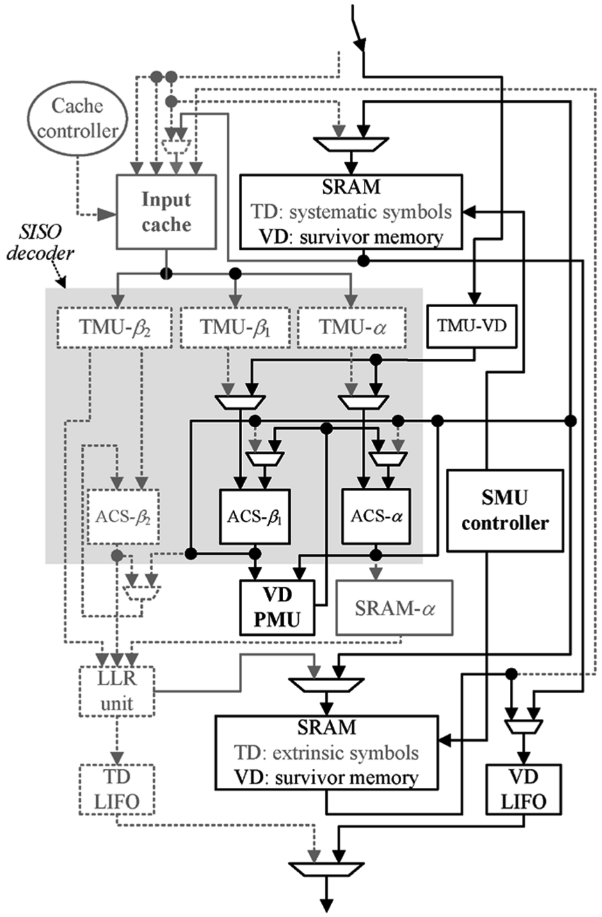


Fig. 7. Decoder in the Viterbi mode.

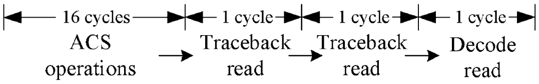


Fig. 8. Timing diagram of the Viterbi decoder.

IV. Chip Implementation

fter fixed-point analysis [15], we provide the fixed representations of the turbo decoder in Table I; the notation $n_i.n_f$ indicates the symbol is $(n_i + n_f)$ bits with n_i integer bits and n_f fractional bits. Moreover, in the VD, the word length of PMs and soft input are determined to be 10 and 4 bits, respectively. The BER performance is shown in Fig. 9. The decoder was implemented in the 0.18-μm standard CMOS technology. In the TD mode, the sub-block length $L_{\rm sb}$ is set to 20, and two clock domains are used in the memory and the datapath, respectively. Since the double clock rate provides the memory with higher bandwidth, the single-port memory is sufficient in the proposed design except the cache memory.

The specification report shows the dual-port memory in Fig. 5 is 0.103 mm^2, leading to a 30% area reduction from Fig. 4, whose area is 0.146 mm^2. Two SP-SRAMs of 20 730 words are included in the decoder for the systematic and the extrinsic data. The input and the output

TABLE I
SUMMARY OF FIXED REPRESENTATION IN TURBO DECODING

quantities	Input symbols	L_{in}	α	β	γ
width	6(3.3)	6(4.2)	8(6.2)	8(6.2)	8(6.2)

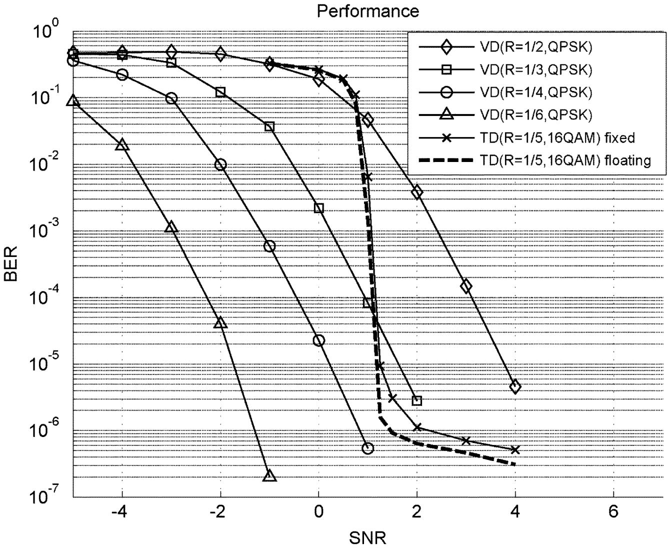


Fig. 9. BER performance of the decoder.

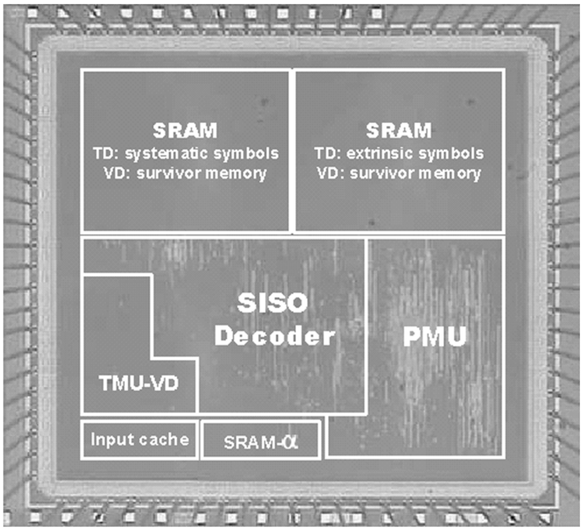


Fig. 10. Microphoto of the decoder chip.

ports are implemented by the time-division multiplexing approach that avoids the use of multiport memories. As compared with DP-SRAM design, the proposed SP-SRAM approach occupies only $1/3$ area with the double clock rate.

In Fig. 10, the chip size is 11.56 mm^2, and the core size is 7.29 mm^2. The total gate count is about 115 k including the PM memory for the Viterbi decoder. Three SP-SRAMs and one DP-SRAM are embedded in the chip with a total size of 251.64 kb. Table II summarizes the chip features where the maximum data rate is obtained from post layout simulation and verified with chip measurement. The power distribution of the major blocks is also illustrated in Fig. 11, where the TD is simulated with the N of 20 730, six decoding iterations, the 16-QAM, and the input SNR of 1 dB, while VD is simulated with the R of $1/6$, the quadrature phase-shift keying (QPSK) modulation, and the SNR of -2 dB.

TABLE II
SUMMARY OF THE DECODER CHIP

Technology	0.18-μm CMOS
Supply voltage	1.8V core / 3.3V IO
Chip size	11.56 mm^2
Embedded SRAM	251.64kb
Supported coding rate	1/5 for turbo mode 1/2,1/3,1/4,1/6 for Viterbi mode
Maximum data rate	4.52Mb/s for turbo mode [1] 5.26Mb/s for Viterbi mode

[1] 6 iterations

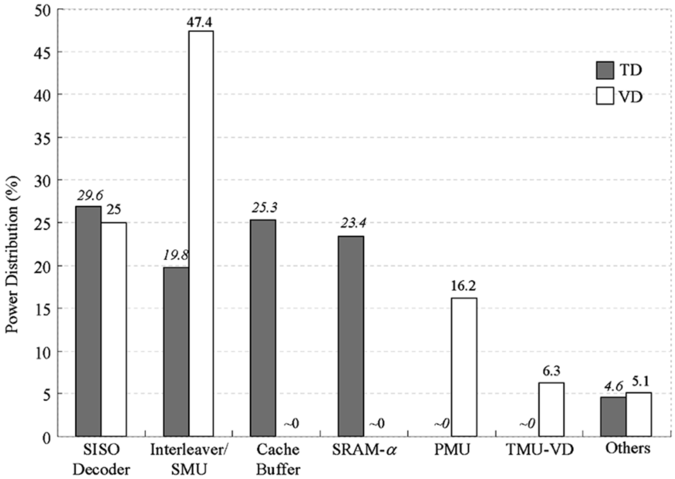


Fig. 11. Power distribution of the major blocks.

TABLE III
POWER CONSUMPTION OF THE DECODER CHIP

Mode	Data rate	Power	SNR
Turbo[1] mode	4.52Mb/s	121 mW	1dB
	3.1Mb/s	83 mW	1dB
	1Mb/s	29.5 mW	1dB
Viterbi[2] mode	5.26Mb/s	116.46 mW	3dB
	1Mb/s	25.1 mW	3dB

[1] R=1/5 and 16-QAM
[2] R=1/2 and QPSK

The chip has been tested at 100 MHz (50 MHz in datapath) under 1.60–1.98 V supply, which can provide the 4.52 Mb/s turbo decoding in six iterations and the 5.26 Mb/s Viterbi decoding. Table III shows the power consumption while decoding turbo and convolutional codes, and Table IV summarizes the differences between the proposed design and other turbo decoder chips. The energy efficiency is defined as the average energy consumed per bit within each decoding iteration (nJ/b/iter). For this decoder with six iterations, the energy efficiency will be $83\ \text{mW}/6 \times 3.1\ \text{Mb/s} = 4.46$ nJ/b/iter.

V. CONCLUSION

In this paper, we present a unified turbo and Viterbi decoder chip with less memory usage and low-power consumption. The memory size is reduced by data scheduling for the interleaver and the single SISO decoder. Furthermore, the power consumption is improved by the efficient memory design and the less data bandwidth for the codeword input. At the 3.1 Mb/s data rate, the power consumption is about

TABLE IV
COMPARISON OF DIFFERENT TURBO DECODER CHIP

	Proposed design	[9]	[10]
Coding rate (R)	1/5	1/3	1/3
Block length	20,730	5,114	5,114
Data rate (Mb/s)	4.52 6 iterations	2.048 10 iterations	24 6 iterations
Technology	0.18-μm	0.18-μm	0.18-μm
Chip core size (mm^2)	7.29	9	14.5[1]
Energy efficiency (nJ/b/iter.)	4.46	14.25	10

[1] Without Viterbi decoder

83 mW in decoding a turbo code with the block length of 20 730. The chip is also designed to work reliably with the wider supply voltage range.

ACKNOWLEDGMENT

The authors would like to thank the National Chip Implementation Center for chip measurement assistance.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. IEEE Int. Conf. Commun.*, 1993, pp. 1064–1070.

[2] S. Benedetto and G. Montorsi, “Unveiling turbo-codes: Some results on parallel concatenated coding schemes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[3] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm with soft-decision outputs and its applications,” in *Proc. IEEE GLOBECOM*, 1989, pp. 47.11–47.17.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[5] P. Robertson, E. Villebrun, and P. Honher, “A comparison of optimal and suboptimal map decoding algorithms operating in the log domain,” in *Proc. IEEE Int. Conf. Commun.*, 1995, pp. 1009–1013.

[6] *Physical Layer Standard for cdma2000 Spread Spectrum Systems*, 3GPP2 Std. C.S0002-C, 2002.

[7] G. Masera, M. Mazza, G. Piccinini, F. Viglione, and M. Zamboni, “Architectural strategies for low-power VLSI turbo decoders,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 3, pp. 279–285, Jun. 2002.

[8] C. Schurgers, F. Catthoor, and M. Engels, “Memory optimization of map turbo decoder algorithms,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 2, pp. 305–312, Apr. 2001.

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

[10] M. Bickerstaff, L. Davis, C. Thomas, D. Garrett, and C. Nicol, “A 24 Mb/s radix-4 log MAP turbo decoder for 3GPP-HSDPA mobile wireless,” in *IEEE Int. Solid-State Circuit Conf. Dig. Tech. Papers*, 2003, pp. 151–484.

[11] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[12] S. A. Barbulescu, “Iterative Decoding of Turbo Codes and Other Concatenated Codes,” Ph.D. dissertation, Univ. South Australia, Adelaide , Australia, 1996.

[13] G. Feygin and P. Gulak, “Architectural tradeoffs for survivor sequence memory management in Viterbi decoders,” *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 425–429, Mar. 1993.

[14] C. C. Lin, Y. H. Shih, H. C. Chang, and C. Y. Lee, “Design of a power-reduction Viterbi decoder for WLAN applications,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1148–1156, Jun. 2005.

[15] Y. Wu, B. D. Woener, and T. K. Blankenship, “Data width requirements in SISO decoding with modulo normalization,” *IEEE Trans. Commun.*, vol. 49, no. 11, pp. 1861–1868, Nov. 2001.

Leakage-Suppressed Clock-Gating Circuit With Zigzag Super Cut-Off CMOS (ZSCCMOS) for Leakage-Dominant Sub-70-nm and Sub-1-V-V_{DD} LSIs

Kyeong-Sik Min, Hun-Dae Choi, H.-Y. Choi,
Hiroshi Kawaguchi, and Takayasu Sakurai

***Abstract*—As a candidate for the clock-gating scheme, Zigzag Super Cut-off CMOS (ZSCCMOS) has proposed to reduce not only the switching power but also the leakage power. Due to its fast wakeup nature, the ZSCCMOS can be best suited to the clock-gating scheme. The wakeup time of the ZSCCMOS is estimated to be 12 times faster than the conventional Super Cut-off CMOS (SCCMOS) in 70-nm process technology. From the measurement of wakeup time in 0.6-μm technology, it is observed to be eight times faster than the conventional scheme. Layout area, power, and delay overhead of the ZSCCMOS are discussed and analyzed in this paper.**

***Index Terms*—Clock-gating circuit, leakage suppression circuit, low-power circuit, Super Cut-off CMOS (SCCMOS), Zigzag Super Cut-off CMOS (ZSCCMOS).**

I. INTRODUCTION

As CMOS technology is scaled down and the supply voltages (V_{DD}s) are further decreased, the threshold voltages (V_{TH}s) should also be scaled down to prevent speed degradation. Decreasing V_{TH} by 0.1 V, however, will increase the subthreshold leakage by more than ten times. Assuming a high-performance device and one million gates in a chip, the chip leakage can reach as much as 40 mA, even in the sleep mode [1]. This large leakage is unacceptable in most portable applications [2].

Of the existing leakage reduction schemes, the Super Cut-off CMOS (SCCMOS) can be used below 1 V V_{DD} without severe speed degradation because the power switch is made with a low-V_{TH} MOSFET. For example, the SCCMOS in [3] can suppress the leakage down to a 1 pA-order per gate when $V_{\mathrm{DD}} = 0.8$ V. Although the SCCMOS successfully suppresses the sleep-mode leakage, the wakeup time is so long that it cannot be used for the active mode. In the active mode, a fast wakeup time is needed to maintain the normal operating speed. The wakeup time of the SCCMOS, amounts up to several clock cycles. In addition, a high -rush current may arise at this transition. The long wakeup time and high rush current make the SCCMOS difficult to use in the active mode where the wakeup occurs frequently. If the SCCMOS is used in the active mode, the several clock cycles of the wakeup process are stolen many times and the overall performance in the active mode is degraded severely.

To overcome the wakeup issues of the SCCMOS, Zigzag-Super-Cut-off CMOS (ZSCCMOS) scheme with a fast wakeup has been proposed and this scheme successfully realizes the clock-gating scheme that saves both the switching and leakage components of power dissipation [4]. The conventional clock gating saves switching power by turning off the local clock whenever the block is not in use. For example, an MPEG-4 decoder chip reportedly saves 72% of the switching

Manuscript received September 15, 2004; revised March 25, 2005 and July 14, 2005. This work was supported in part by the Mirai-Kaitaku project, Japan, and by the Basic Research Program of KOSEF, Korea under Grant R01-2003-000-11639-0.

K.-S. Min, H.-D. Choi, and H.-Y. Choi are with the School of Electrical Engineering, Kookmin University, Seoul 136-702, Korea (e-mail: mks@kookmin.ac.kr).

H. Kawaguchi was with the Institute of Industrial Science, Center for Collaborative Research, University of Tokyo, Tokyo 153-8505, Japan. He is now with the Department of Computer Systems Engineering, Kobe University, Kobe 657-8501, Japan.

T. Sakurai is with the Institute of Industrial Science, Center for Collaborative Research, University of Tokyo, Tokyo 153-8505, Japan.

Digital Object Identifier 10.1109/TVLSI.2006.874378