LOW-POWER BIT-SERIAL VITERBI DECODER FOR NEXT GENERATION WIDE-BAND CDMA SYSTEMS

Hiroshi Suzuki

Kawasaki Steel Corporation
LSI Division
1-5 B5 Nakase Mihama-ku, Chiba, Japan

Yun-Nan Chang *Keshab K. Parhi*

Department of Electrical and
Computer Engineering
University of Minnesota
Minneapolis, MN 55455, USA
{suzuki, ynchang, parhi}@ece.umn.edu

ABSTRACT

This paper presents a low-power bit-serial Viterbi decoder chip with the coding rate $r = 1/3$ and the constraint length $K = 9$ (256 states). This chip has been implemented using 0.5μm three-layer metal CMOS technology and is targeted for high speed convolutional decoding for next generation wireless applications such as wide-band CDMA mobile systems and wireless ATM LANs. The chip is expected to operate at 20Mbps under 3.3V and at 2Mbps under 1.8V. The Add-Compare-Select (ACS) units have been designed using *bit-serial arithmetic*, which has made it feasible to execute 256 ACS operations in parallel. For trace-back operations, we have developed a novel power-efficient trace-back scheme and an *application-specific memory*, which was designed considering that 256 bits should be written simultaneously for write operations but only one bit needs to be accessed for read operations. We have estimated that the chip dissipates only 10mW at 2Mbps operation under 1.8V.

1. INTRODUCTION

Many Viterbi decoders have been developed for various applications [1] [2] [3] [4]. These Viterbi decoders are divided into two classes. The decoders in [1] [2] are mainly targeted for magnetic storage devices and their goal is to achieve very high speed operations for a small number of states. On the other hand, the decoders in [3] [4] have a long constraint length and a large number of states, but their target speed is moderate. Their target application is IS-95 CDMA mobile system, where a convolutional code with the constraint length $K = 9$ is used. In CDMA systems, it is necessary to reduce the required SNR (Signal Noise Ratio) at a given BER (Bit Error Rate) since one user's transmission signal is considered as noise by the other users and high transmission signal level leads to the decrease of the channel capacity. This is the reason why such a long constraint code with strong error correction capability is used in CDMA systems. In [3] [4], ACS operations are executed serially using a small number of ACS units. This approach is possible because the required speed is moderate. Recently, the demand for higher speed CDMA systems is increasing and new standards are emerging [5]. In [5], 2Mbps is defined as the maximum speed, which is more than one hundred times as fast as the transmission speed of the current CDMA mobile standard. Designing high speed Viterbi decoders with a large number of states is quite challenging. It is necessary to finish all 256 ACS operations within a specified bit transmission time. Furthermore, the survivor path trace-back operation is another challenging part. It is difficult to apply simple trace-back schemes described in [3] [4] to high speed decoders. Some special techniques are required to achieve high speed operation as described in [6] [7].

In this paper, we present the design of a low-power Viterbi decoder with a long constraint length which is targeted for next generation wireless applications. We have adopted *bit-serial arithmetic* for the ACS operations. The bit-serial arithmetic leads to area-efficient datapath units and requires low power dissipation due to less glitching activities [8]. The bit-serial approach has made it possible to obtain an area and power efficient ACS architecture. Regarding the trace-back block, we have developed a novel trace-back technique which is simpler than the techniques described in [6] [7]. Our trace-back technique is very power efficient due to the use of *application-specific memories*. The readers are assumed to be familiar with the Viterbi algorithm. For details on the Viterbi algorithm, see [9].

2. ARCHITECTURE OVERVIEW

The specification of the Viterbi decoder we have designed is summarized in Table 1. The block diagram of the chip is shown in Fig. 1. In this chip, 256 bit-serial ACS units are placed in parallel and each ACS unit includes state metrics storage. All state metrics connections between the ACS units and all branch metrics connections between the branch metrics unit (BMU) and the ACS units are done with a single bit wire, resulting in a drastic reduction of the routing area compared to the bit-parallel approach. Regarding the trace-back block, a 256×48 bit memory is required for the survivor path length of 48. The size of the memory used in the chip is

Table 1: Viterbi Decoder Specification

Constraint Length	$K = 9$ (256 states)
Coding Rate	$r = 1/3$
Generator Polynomials	$G0 = 557$, $G1 = 663$, $G2 = 771$
Survivor Path Length	$D = 48$
Input	3bits Soft Decision
Target Speed	20Mbps (3.3V), 2Mbps (1.8V)

This work was carried out at the University of Minnesota, and has been supported by DARPA under contract number DA/DABT63-96-C-0050.

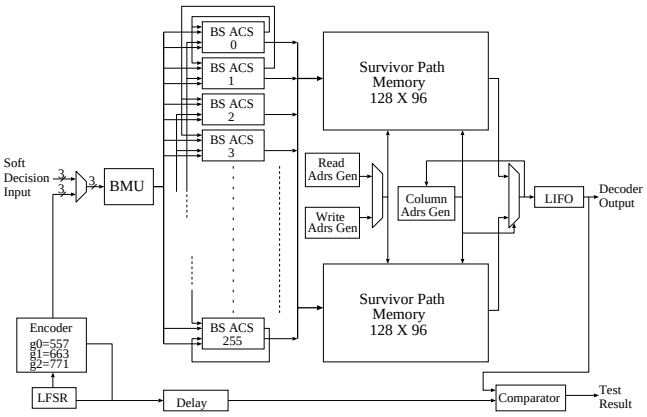

Figure 1: Bit-Serial Viterbi Decoder Chip Diagram

twice the required size. This extra cost leads to a high-speed traceback operation and lower power consumption (see Section 4).

3. BIT-SERIAL ACS OPERATION

The core part of the Viterbi algorithm is the ACS operation, which decides the shortest path for a state among all possible paths. For the rate $r = 1/3$ decoder, only two paths go into one state and we can obtain a single decision bit with a single ACS operation. The ACS unit calculates the state metrics for the next trellis stage as well. The ACS operation is described by

$$SM^0_{n+1} = min\left(SM^0_n + BM^0_n,\ SM^1_n + BM^1_n\right)$$

where $SM^i_n (i = 0, 1)$ are state metrics at time n and $BM^i_n (i = 0, 1)$ are branch metrics at time n. Fig. 2 shows a diagram of the ACS unit. The word length of the state metrics and the branch metrics are 8-bit and 5-bit, respectively. The decision bits are obtained as a result of the comparison. We have adopted the bit-serial arithmetic for the ACS units, which made it possible to execute 256 ACS operation in parallel. The same speed can be achieved us-

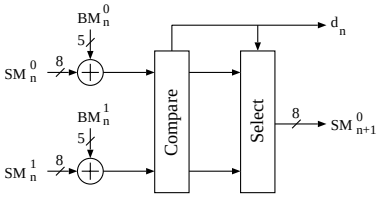

Figure 2: ACS unit diagram

ing fewer bit-parallel ACS units, for example, 32 or 64 bit-parallel ACS units repeatedly. In this approach, however, the state metrics storage needs to be implemented using actual memories instead of registers as described in [10]. It results in the significant area and power overhead of control circuits such as selectors and a state metrics memory address generator. We decided to use the bit-serial approach considering these issues and the following advantages and disadvantages. The advantages of the bit-serial approach are:

- The performance obtained from the bit-serial approach is suitable for the target speed (20Mbps@3.3V, 2Mbps@1.8V).
- Glitching activities can be reduced.
- Routing area consumed for state metrics and branch metrics routing is drastically reduced.

On the other hand, the disadvantages of the bit-serial approach are:

- Extra storage elements are required to store the state metrics which will not be used after a decision is obtained.
- Registers are needed for intermediate carries for the bit-serial additions and subtractions.

Since the second one is inherent in bit-serial arithmetic, it is impossible to avoid this problem. As for the first one, we have designed area-efficient storage elements based on the ring-type FIFO architecture and have overcome this disadvantage successfully.

3.1. Bit-Serial ACS Units

The architecture of the bit-serial ACS unit is depicted in Fig. 3. One of two FIFOs is used as a storage element for the state metrics depending on the decision bit. The circuit for the FIFO is basicly the same as one described in [11]. As mentioned above, an extra set of storage elements are required if we choose the bit-serial approach. Using the FIFO based structure shown in Fig. 3, we succeeded in reducing the overhead down to 17% of the whole area of the ACS unit. It is possible to implement the storage elements with simple

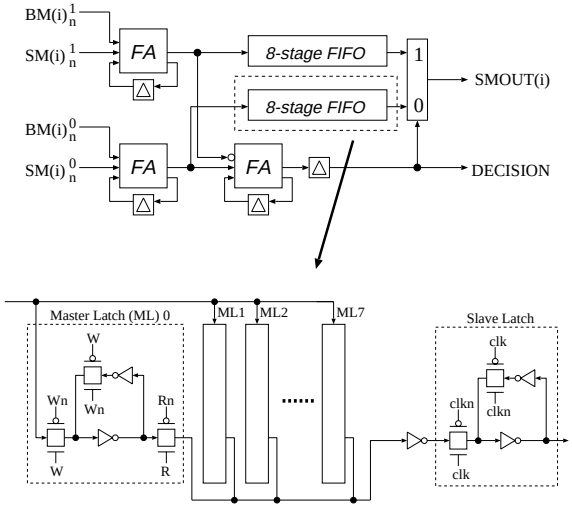

Figure 3: Bit-serial ACS unit

shift registers. It does not require any control signals such as write enables or read enables and incoming data is just transfered from one stage to another. Shift register based architectures are much simpler, but these dissipate more power than the ring-type architecture. In the ring-type architecture, the bit position where incoming data should be written is specified by a write pointer. In the shift register based one, however, incoming data are always written to the same register, namely, the register at the first stage and at the same time old data need to be transfered to latter stages. This leads to a significant increase in the switching activities in flip-flops resulting in high power dissipation. According to our power estimation of these two architectures, the shift register based circuit consumes 35% more power than the ring-type one.

3.2. State Metric Routing Strategy

The interconnection of state metrics between the ACS units is one of the key issues in the implementation of Viterbi decoder with a large number of states. Direct implementation without any routing strategy consideration could result in tremendous routing overhead even larger than the ACS units themselves. The complexity of state metrics routing is mainly affected by the placement of the ACS units; therefore, a systematic approach based on searching the Hamilton cycle in the deBruijn graph has been proposed in [12] to find an area-efficient ACS topology. Here the ACS units are arranged in pairs and all ACS pairs can be organized in a two column topology such that one of the two outgoing state metrics of each ACS pair will be connected to the adjacent pair. This approach is not suitable for our design because of the large aspect ratio of the two column topology. In addition, since the ACS units are not placed in a regular order, the routing of decision bits between ACS units and the path memory will become very complicated and cannot be neglected especially in the bit-serial ACS implementation. Therefore, a new ACS topology which is not only area-efficient but also simpler in the layout implementation is used in our design. As shown in Fig. 4, 256 ACS units are divided into 8 clusters and most of the outgoing state metrics from the clusters simply go into the neighboring clusters. Considering the complete ACS module excluding the decision bit routing, our layout implementation shows the state metric routing only occupied about 25% of total area of the ACS module.

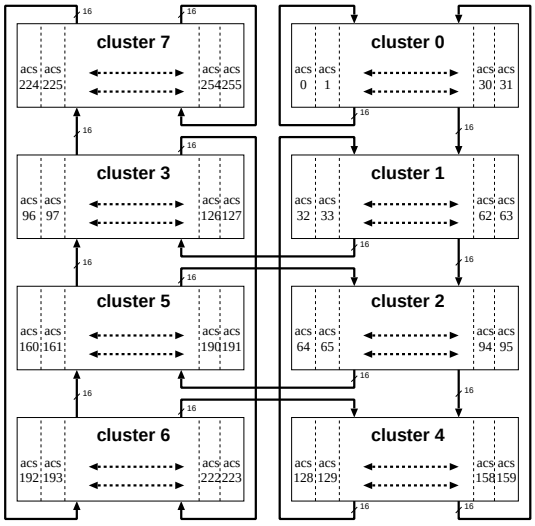

Figure 4: Floor plan of the ACS module.

4. TRACE BACK UNIT

The input sequence is decoded using a recursive procedure known as trace-back. Let us define S_n as an arbitrary state at time n which is represented with the 8-bit binary form. Each bit in S_n corresponds to consecutive input data to the shift register in the convolutional encoder and the MSB is considered as the oldest input value. 256 decision bits are read from the survivor path memory using time n as a read pointer. The decision bit for S_n, d_n^S is selected using S_n as a select signal. The previous state is given by $S_{n-1} = d_n^S\,(S_n >> 1)$. After this process is repeated to the time point which corresponds to the survivor path length, a single decoded bit is obtained. It should be noticed that only one bit from 256 decision bits is required for the trace-back recursions. This observation is exploited to design a power-efficient survivor path memory (see Section 4.2).

4.1. Trace Back Strategy

For the trace back operation with $D = 48$ (survivor path length), a 256×48 bit memory is at least required. A single decoded bit is obtained after 48 read operations if a minimum size memory is used. This operation is applicable only to low-speed decoders as described in [3] [4], but it requires almost 100MHz clock for memory accesses for 2Mbps decoding and cannot be used for a decoding rate of 20Mbps. Much research has been done for high speed trace-back operations [6] [7] . We have adopted a trace-back strategy which is similar to the one described in [6], but is much simpler. In our approach, only one single-port memory and a single read pointer are needed. The approach in [6], however, requires at least two memories and multiple read pointers. Two memories are placed in our chip as shown in Fig. 1, but we divided one memory into two pieces for the purpose of reducing power. Now we explain our trace back strategy. The memory size required in our approach is twice as large as the minimum memory size (256×48). The trace-back strategy is depicted in Fig. 5, where W, T and D represent "WRITE", "TRACE BACK" and "DECODE", respectively. During "WRITE" operation, 256 decision bits are written

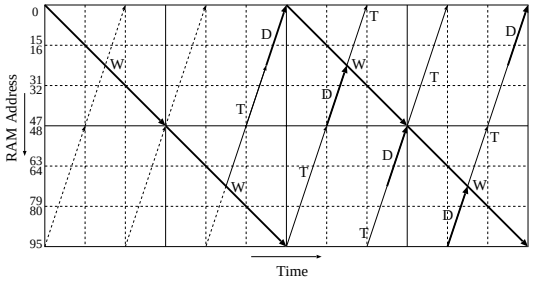

Figure 5: Trace Back Operation

to survivor path memories simultaneously. Both "TRACE BACK" and "DECODE" are read operations, but only "DECODE" gives decoded outputs. After 48 "TRACE BACK" operations, 24 decoded bits are obtained consecutively. Compared to the approach in [4], the number of read operations required to obtain a single decoded bit was reduced from 49 to 3 and it results in dramatic power reduction of the trace-back block. As shown in Fig. 5, two separate pointers, namely, a read pointer and a write pointer are required and the speed of the read pointer should be three times as fast as that of the write pointer. This operation was implemented with single-port memories using a time-multiplexed access method. It should be noted that the decoded data generated by the trace-back recursion process are reverse in order. Hence, a LIFO (Last-In-First-Out memory) needs to be placed after the trace-back block as shown in Fig. 1.

4.2. Survivor Path Memory Architecture

We have designed an application-specific memory for the survivor path memories in order to reduce the power dissipation. In the trace back operation, all bits are accessed during decision bit write operations, but only one bit needs to be accessed during read operations as explained above. This feature of read operations has been exploited to reduce power consumption. First, The total memory

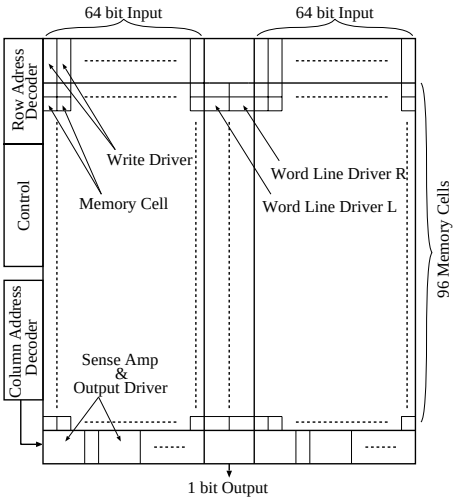

Figure 6: 128×96 Survivor Path Memory

(256×96 bits) has been divided into two blocks. One of these blocks is activated during read operations according to the result of the previous read operation. In each memory, word lines are divided into two parts and only half of memory cells become active. Furthermore, only one sense amp and only one output driver are activated depending on which bit needs to be read. The architecture of the survivor path memory is shown in Fig. 6. In this architecture, 128 write drivers are located on the top and word line drivers are placed in the middle dividing bit lines into two groups. Column address is effective only for the read operation and selects a sense amp and a output driver which should be activated. We have estimated that the trace-back memories dissipate about 2mW for 2Mbps operation under 1.8V.

5. SUMMARY

The chip layout is shown in Fig. 7. The chip has been fabricated using HP 0.5μm 3-layer metal CMOS technology. The chip size is 3.19mm×4.88mm and the core size is 2.46mm×4.17mm. It consists of 200K transistors. More than 45% of these are used for two survivor path memories. We have estimated the power dissipation of the core with the Hspice simulation. The core dissipates 10mW at 2Mbps under 1.8V and 72μW at 14.4Kbps under 1.8V. Compared to the result in [4] (240μW at 14.4Kbps), power dissipation of the core is one-third less than the previous chips. The chip is expected to operate at 20Mbps under 3.3V as well, which indicates that the core can be used for very high speed wireless applications such as wireless ATM LANs.

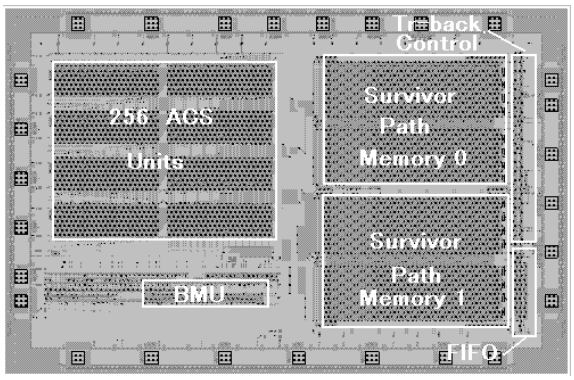

Figure 7: Chip layout

6. REFERENCES

[1] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder", *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1877–1885, Dec. 1992.

[2] A. K. Yeung and J. Rabaey, "A 210-Mb/s, radix-4 bit-level pipelined Viterbi decoder", *ISSCC Digest of Technical Papers*, Feb. 1995.

[3] J. K. Hinderling *et al.*, "CDMA mobile station modem ASIC", *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 253–260, March 1993.

[4] I. Kang and A. N. Willson Jr., "Low-power Viterbi decoder for CDMA mobile terminals", *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 473–482, March 1998.

[5] "Japan's proposal for candidate radio transmisson technology pn IMT-2000:W-CDMA", *Association of Radio Industries and Businesses (ARIB), Japan*, June 1998.

[6] R. Cyper and C. B. Shung, "Generalized trace-back techniques for survivor memory management in the Viterbi algorithm", *Proc. GLOBECOM*, vol. 2, pp. 1318–1322, Dec. 1990.

[7] H. A. Bustamente *et al.*, "Stanford telecom VLSI design of convolutional decoder", *Proc. MILCOM*, vol. 1, pp. 171–178, Oct. 1989.

[8] R. I. Hartley and K. K. Parhi, *Digit-Serial Computation*, Kluwer Academic, Boston, MA, 1995.

[9] G. D. Forney Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 268–278, March 1973.

[10] C. M. Rader, "Memory management in a Viterbi decoder", *IEEE Transactions on Communications*, vol. 29, pp. 1399–1401, Sept. 1981.

[11] C. J. Nicol *et al.*, "A low-power 128-tap digital adaptive equalizer for broadband modems", *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 1777–1789, Nov. 1997.

[12] J. Sparso *et al.*, "An area-efficient topology for VLSI implementation of Viterbi decoders and shuffle-exchange type structures", *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 90–97, Feb. 1991.