Two Bit-Level Pipelined Viterbi Decoder for High-Performance UWB Applications

Yong-Je Goo and Hanho Lee
School of Information and Communication Engineering, Inha University
253 Yonghyun-Dong, Nam-Gu, Incheon, 402-751, Korea
Email: ggooyong@inhaian.net, hhlee@inha.ac.kr

Abstract—This paper presents a high-speed low-complexity two bit-level pipelined Viterbi decoder architecture for MB-OFDM UWB systems. As the add-compare-select unit (ACSU) is the main bottleneck of the Viterbi decoder, this paper proposes two bit-level pipelined MSB-first ACSU, which is based on 2-step look-ahead techniques, to reduce a critical path of the ACSU. The proposed ACSU architecture requires approximately 12% fewer gate counts and 9% faster speed than the conventional MSB-first ACSU. The proposed Viterbi decoder was implemented with 0.18-μm CMOS standard cell technology in a supply voltage of 1.8V. It operates at a clock frequency of 870 MHz and has a throughput of 1.74 Gb/s.

I. INTRODUCTION

The Viterbi algorithm (VA), which is widely used for channel decoding in digital telecommunications, was first introduced as a method for convolutional decoding in 1967 [1]. Generally, a Viterbi decoder consists of three basic computation units: the branch metric unit (BMU), the add-compare-select unit (ACSU), and the TraceBack unit (TBU), as shown in Fig. 1. The BMU calculates the branch metrics by the hamming distance or the Euclidean distance, and the ACSU calculates a summation of the branch metric from the BMU and previous state metrics, which are called the path metrics. After this summation, the value of each state is updated and then the survivor path is chosen by comparing path metrics [1]. The TBU processes the decisions made in the BMU and ACSU, and outputs the decoded data. The feedback loop of the ACSU is a major critical path for the Viterbi decoder. There are several techniques to implement a high speed Viterbi decoder. In [2], the M-step look-ahead technique, a well-known high speed algorithm speed algorithm for elimination of feedback operation, is utilized. Generally, as \( M \) is increased, the power exponentially. Thus, \( M \) is limited to 2. Fig. 2 shows trellis diagrams for the case of 4 states. Fig. 2(a) is a general trellis diagram of a Viterbi decoder in which each state has two branches connected to the other states. The trellis diagram of the Viterbi decoder is iterated as same paths in every time, as shown in Fig. 2(a), and thus it can be modified to a 2-step look-ahead structure, as shown in Fig. 3(b) [2]. The least significant bit (LSB) first computation is available for accumulation, similar to the summation, but the most significant bit (MSB) first computation is suitable for comparing and selecting operations in the ACSU. The ACSU structure combining MSB-first compare-select with carry propagation was proposed based on redundant number representation in [3]. The CSAU structure was proposed to reduce the critical path of the ACS operation [9]. However, when the ACSU is constructed for the radix-4 structure, the area of the CSAU is increased exponentially and the critical path of the CSAU is increased by the total bits of the adder.

In this paper, a high-speed two-bit level pipelined MSB-first ACSU is proposed with the aim of reducing the hardware complexity and improving the clock frequency in...
the Viterbi decoder. This design does not requireadder computation and code converter delay, which are required in the critical path of the conventional bit-level pipelined ACSU [3].

II. TWO BIT-LEVEL PIPELINED VITERBI DECODER

A. Depunctured Unit

The punctured Viterbi decoder is used for diverse code rates using one Viterbi decoder. To implement the punctured Viterbi decoder, a depunctured unit is added in the front of the BMU. The depunctured unit is implemented for code rates of 1/2, 1/3, 3/4, and 5/8, and the basis code rate of the Viterbi decoder is 1/3. The depunctured unit takes the input signals and the code rate mode signal, and the input signal is then changed to an appropriate signal by a depunctured metric. The outputs of the depunctured unit, which are appropriately changed to 2 input symbols of the Viterbi decoder by using 2-step look-ahead techniques, are passed to the BMU.

B. Branch Metric Unit

The integers for branch metrics are used instead of log likelihoods. This integer method has a simple structure and offers negligible performance reduction for 8-level quantization [5]. Branch metrics for the 2-step look-ahead trellis are generated by combining branch metrics of successive iterations underlying two number of the radix-2 trellis structure. For the each iteration of the radix-2 trellis from n to n+1, three 3-bits soft-decision inputs can be constructed Branch metric according to input soft bits. as \( G_3, G_2, G_1 \). As shown in Table I, each input \( G \) can be implemented as follows:

\[
\{BM_i[2], BM_i[1], BM_i[0]\} = \{G[2], G[1], G[0]\}
\]

\[
BM : \text{branch metric}
\]

In addition, each of the three 3-bits soft-decision inputs can be organized to logically form eight possible branch metrics, as \( BM_{000}, BM_{001} \ldots BM_{111} \). 64 possible 2-step look-ahead branch metrics for iteration from n to n+2 are formed from the radix-2 metrics \((G_3, G_2, G_1)\times(G_3, G_2, G_1)\) for iterations n and n+1. In this design, the radix-4 metrics are centrally calculated and then globally distributed to the ACSUs. As shown in Table I, the branch metric is constructed as

\[
BM_{000, 000} = BM_{000}^n + BM_{001}^{n+1}
\]

\[
BM_{000, 001} = BM_{000}^n + BM_{001}^{n+1}
\]

\[
\vdots
\]

\[
BM_{111, 110} = BM_{111}^n + BM_{110}^{n+1}
\]

\[
BM_{111, 111} = BM_{111}^n + BM_{111}^{n+1}
\]

Because the constraint length of the convolution encoder is 7, the total number of the states is 64 on the UWB system, as given by \(2^{7-1}\).

---

**TABLE I. BRANCH METRIC ACCORDING TO INPUT SOFT BITS.**

<table>
<thead>
<tr>
<th>(G)</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM_{00}</td>
<td>011(3)</td>
<td>010(2)</td>
<td>001(1)</td>
<td>000(0)</td>
<td>111(7)</td>
<td>110(6)</td>
<td>101(5)</td>
<td>100(4)</td>
</tr>
<tr>
<td>BM_{10}</td>
<td>100(4)</td>
<td>101(5)</td>
<td>110(6)</td>
<td>111(7)</td>
<td>000(0)</td>
<td>001(1)</td>
<td>010(2)</td>
<td>011(3)</td>
</tr>
</tbody>
</table>

---

**Fig. 4.** Block diagram of the proposed 2-bit level pipelined Viterbi decoder.
C. Proposed Add-Compare-Select Unit

The recursive state metric update results in unbounded word growth due to the addition of branch metrics, which are always nonnegative. Generally, the proved modular arithmetic approach is adopted to avoid normalization [5]. For the 64-state 2-step look-ahead decoder implementation, the required state metric precision is 9 bits, as given by

$$S_{\text{bits}} = \log_2(\lambda_{\max} + \lambda_{\max}^2 + 2) + 1$$

where $\lambda_{\max} = 21$, and $\lambda_{\max} = 126$.

The proposed MSB-first ACSU is pipelined in 2 bit-levels to reduce the critical path of the ACSU, as shown in Fig. 5. The proposed MSB-first ACSU can be divided into 3 basic function blocks, which are the 2-bits adders, the maximum position (MP) blocks, and the maximum selection (MS) blocks. To implement the two-bit level pipelined MSB-first ACSU, it is necessary to consider the influence of the carries in the 2-bits full adders, which can affect the decision of the maximum state value in each state at time $n+1$, $n+2$, $n+3$.

The MP block is designed to defend against the influence of the carries, which occur at time $n+1$, $n+2$, $n+3$, as shown in Fig. 7. The MP block has three sub-blocks: the partial decision (PD) block, the maximum-position-detection (MPD) block, and the maximum-position-selection (MPS) block. The PD block is constructed to find which state has 1 less value than the maximum value ($S^\text{p}^n$, $S^\text{PS}$) among the four branch values $\{S^\text{p}^n[1], S^\text{p}^n[0], S^\text{PS}^n[1], S^\text{PS}^n[0]\}$ at time $n$. If one of the branch values, which are calculated at time $n$, is a value 1 less than the maximum value that is calculated at time $n$, the output of the PD block is 1.

The MPD block finds which state has the same value as the maximum value ($S^\text{MPD}^n[1], S^\text{MPD}^n[0]$) at time $n$ and checks the carries at time $n+1$ by considering the influence of the carries ($C^s_{a[1]}$, $C^s_{a[1]}$, $C^s_{a[1]}$, $C^s_{a[1]}$). The MPS block is constructed to decide which branch value is the maximum branch value in each state. The MPS block receives the inputs from the result of the PD block, MPD block, and the carries at time $n+1$. After the operation of the ACSU is finished, the output of the MPS in the least significant bits (LSB) is handed over to the TBU to decide the survivor path at each time and store the information of the survivor path in the survivor path memory in the TBU.

D. TraceBack Unit

The register exchange algorithm and the TraceBack algorithm are representative methods to decide the decoded output. As the TraceBack algorithm uses less power than the register exchange algorithm, it is more suitable [8]. To obtain the decoded output of the Viterbi decoder, the TBU first has to decide the maximum state among the 64 states, which is the initial state for the TraceBack operation. As the TBU invokes too much delay time to compute the maximum states among the 64 states at one time for the TraceBack operation, 64-states are separated to 16 groups per 4 states.
The TBU finds the maximum states in each group at time \( n \). The outputs of the 16 groups are then separated to these 4 groups and are calculated to find the maximum states in each group at time \( n+1 \). Finally, the last 4-states that are selected to maximum state at time \( n+1 \) are compared to decide the initial state for the TraceBack operation.

### III. IMPLEMENTATION AND COMPARISON

The proposed Viterbi decoder was modeled in Verilog HDL and simulated to verify its functionality. After complete verification of the design functionality, it was then synthesized using appropriate time and area constraints. Both simulation and synthesis steps were carried out using SYNOPSYS design tools and 0.18-\( \mu m \) CMOS technology optimized for a 1.8V supply voltage. Table II presents a comparison of the critical path with the conventional ACSU [3][9]. The critical path of the proposed 2-bit level pipelined ACSU requires fewer XNOR, AND gates compared to the conventional bit-level pipelined ACSU [3], and the critical path of the CSAU [9] requires more AND, OR, XOR gates compared with the proposed ACSU.

Table III shows the hardware complexity for the proposed two-bit level pipelined ACSU and the conventional bit-level pipelined ACSU. The proposed ACSU in each state requires 7% fewer gates than does the conventional bit-level pipelined ACSU. When the ACSU is implemented for 64 states, the number of gates for the proposed ACSU is reduced by about 12%, as shown in Table III. The total number of gates is 148,101 from the synthesized results excluding the survivor path memories. The proposed Viterbi decoder operates at a clock frequency of 870 MHz and has a throughput of 1.74 Gb/s.

### IV. CONCLUSION

This paper presents a two-bit level pipelined Viterbi decoder for high-performance UWB applications. A two-bit level pipelined ACSU is proposed to reduce the area and the critical path delay of the Viterbi decoder. As a result, the Viterbi decoder using the proposed pipelined ACSU has lower hardware complexity compared to a previously reported Viterbi decoder based on the conventional bit-level pipelined ACSU. Finally, the performance results show that the data throughput is as high as 1.74 Gb/s while requiring small hardware complexity.

### V. ACKNOWLEDGMENT

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (IITA-2006-C1090-0603-0019).

### REFERENCES


