

Implementation of an Efficient Two-Step SOVA Turbo Decoder for Wireless Communication Systems

J. H. Han¹, A. T. Erdogan^{1,2}, T. Arslan^{1,2}

¹University of Edinburgh, School of Engineering and Electronics
Edinburgh, EH9 3JL, Scotland, United Kingdom

²Institute of System Level Integration, The ALBA campus
Livingston, EH54 7EG, Scotland, United Kingdom

j.han@ed.ac.uk, Ahmet.Erdogan@ee.ed.ac.uk, Tughrul.Arslan@ee.ed.ac.uk

Abstract—The authors present a soft-input soft-output (SISO) turbo decoder based on the two-step soft-output Viterbi algorithm (SOVA). The turbo decoder has been implemented with the trace-back algorithm (TBA). It achieves power and area reduction by employing an additional transition metric unit (TMU) for the generation of reliability values, instead of using a FIFO block for retrieving the previously stored reliability values in the survivor memory unit (SMU). Simulation results are provided for bit error rate (BER) performance using constraint lengths of $K=3, 4$, and 5 . Implementation results for $K=5$ are also provided showing 46% area and 15% power savings compared to a conventional SOVA decoder.

Key words—Turbo decoder, low power, soft output Viterbi algorithm, trace back algorithm

I. INTRODUCTION

The demand of turbo codes for wireless communication systems has been increasing since they were first introduced by Berrou et. al. in the early 1990s [1]. Various systems such as 3GPP, HSDPA and WiMAX have already adopted turbo codes in their standards due to their large coding gain. In [2], it has also been shown that turbo codes can be applied to other wireless communication systems used for satellite and deep space applications.

Two algorithms, maximum *a posteriori* (MAP) [1] and soft-output Viterbi algorithm (SOVA) [3], are widely used for turbo decoder implementations. While MAP based turbo decoders generally provide better performance in terms of bit error rate (BER), their complexity and latency is very high. On the other hand, SOVA based turbo decoders can be implemented with high throughput and less complexity. However, their BER performance is less than MAP decoders. Although MAP decoders can address the latency problem by using the sliding window (SW) method [4], their complexity is still very high.

Many researchers have studied the SOVA decoder algorithm and architecture levels in order to improve its

performance in terms of BER, power, and area. In [5], a high throughput SOVA decoder with the register exchange algorithm (REA) has been implemented. A low power SOVA decoder using a systolic array in its trace back unit has been introduced in [6].

In this paper, we present a soft-input soft-output (SISO) turbo decoder based on the two-step SOVA [7]. The two-step SOVA, which separates the decoding process into two processes called survivor and update, allows for the implementation of the turbo decoder with less complexity compared to the original SOVA. Low power is one of the most critical parameters for decoders targeting mobile wireless communication systems. It is well known that while REA is more suitable for high throughput applications, trace back algorithm (TBA) can be implemented with low power consumption. Therefore, our two-step SOVA turbo decoder has been implemented with TBA. Furthermore, our turbo decoder employs an additional transition metric unit (TMU) for the generation of reliability values, instead of using a FIFO block for retrieving the previously stored reliability values in the survivor memory unit (SMU). We have also investigated the BER performance of our turbo decoder for constraint lengths $K=3, 4$, and 5 in order to verify its functionality. In order to evaluate the area and power efficiency of our turbo decoder, it has been implemented in Verilog HDL and synthesized using a 0.18 μ m standard cell library. The results show that, for constraint length $K=5$, our decoder achieves 46% area and 15% power savings compared to a conventional SOVA turbo decoder.

II. SOFT OUTPUT VITERBI ALGORITHM FOR TURBO CODES

In this section, we briefly described the decoding process of the two-step SOVA turbo decoder. Fig. 1 shows an example of a trellis diagram for two-step SOVA decoding process for constraint length $K=3$. As can be seen, the decoding process is divided into survivor and update processes. During the survivor process from time k to time $k-D$ (D is the decoding length), the SOVA finds the maximum likelihood (ML) state which all survivor paths

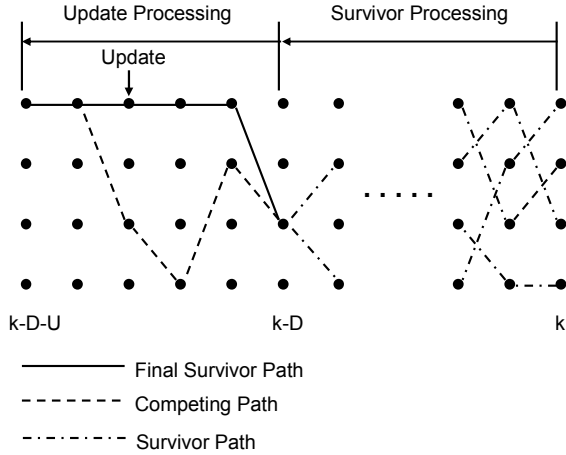


Figure 1. Example trellis diagram for two-step SOVA decoding.

merge at time $k-D$ and computes its reliability value. The update process starts at time $k-D$ and ends at time $k-D-U$ (U is the update length). During the update process two decision bits, survivor path decision bit $u_s(k)$ and competing path decision bit $u_c(k)$, and the reliability values are compared and updated for each state along side the survivor and competing paths. The update rule relates to the decision bits and the reliability value which is given below [3]:

for $j=k-D, k-D-1, \dots, k-U$

if $u_s(j) \neq u_c(j)$

$$L_r(j) \leq \min(L_r(j), \Delta_p(k))$$

else

$$L_r(j) \leq L_r(j)$$

where $L_r(j)$ and $\Delta_p(k)$ are the temporal reliability and the computed reliability values obtained by the TMU. The final reliability value $L_{fr}(k)$ and hard decision bit $u(k)$ generated by the updating process represent the log likelihood ratio (LLR):

$$L_{lr}(k) = u(k)L_{fr}(k) \quad (1)$$

Consequently, the LLR values have hard decision bits information and reliability values. In a turbo decoder, the LLR is used for the next decoding process after subtracting the two soft input values. The following represents the extrinsic information in a turbo decoder:

$$L_{2ex}(k) = L_{lr1}(k) - L_c \cdot x_k - L'_{1ex}(k) \quad (2)$$

where L_c and x_k are the reliability value of the channel and the input data of systematic bits, respectively. Equation (2) shows the relationship between the soft output LLR and the extrinsic information. In [8], the relationship between LLR values and extrinsic information as well as SNR-related issues is explained in more detail.

III. IMPLEMENTATION

A block diagram of our turbo SISO decoder is shown in Fig. 2. The two-step SOVA turbo decoder is based on TBA and consists of TMU, SMU, and UPU (Update Process Unit). In the conventional implementation, a FIFO block is incorporated in SMU to store the reliability values generated by TMU [4]. However, SMU implemented in this paper incorporates another transition metric unit (TMU2) instead of using a FIFO block. In [9], a SOVA decoder without a memory block to store the reliability values has been suggested. However, a memory block is still required to store the input data for computing the reliability values. This memory block can be removed using a dual read port memory block for the input buffer, as shown in Fig 2. This leads to more savings in area usage and power consumption. The following sub-sections provide more information on hardware implementation of the main units.

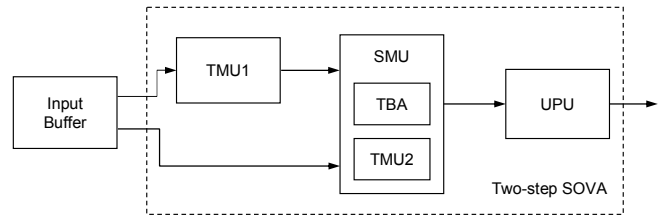


Figure 2. A block diagram of proposed SOVA decoders.

A. TMU

The TMU architecture and its main sub component, add-compare-select (ACS), are shown in Fig. 3 (a) and (b) for constraint length $K=3$. In Fig. 3 (a), two 3-input adders are used to compute the branch metric values (bm_0, bm_1, bm_2 , and bm_3). These values are then input to four parallel ACS modules to compute the decision bits (d) and the reliability values (r_0, r_1, r_2 , and r_3). The decision bits are used to find

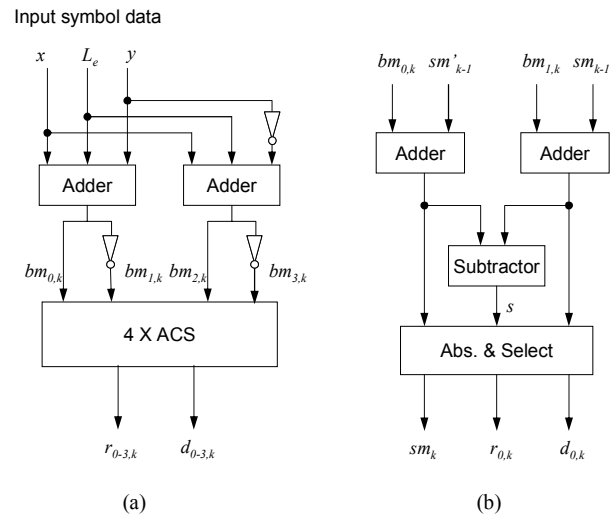


Figure 3. Architectures of (a) TMU and (b) ACS as constraint length $K=3$.

decoding bit by trace back algorithm in the SMU and the reliability values are delayed until finding the decoding bit and used by UPU to generate the LLR values. The ACS module in Fig. 3 (b) consists of two Adders, one Subtractor, and one Abs. & Select units. Each adder is used to compute a state metric value by adding a path metric value to a previous state metric value. The two state metric values are then compared through the Subtractor. One of these state metric values is selected as the next state metric value (sm_k) based on the decision bit (d), which in turn is determined by the most significant bit (MSB) of the Subtractor output. On the other hand, the reliability value (r_0) is determined by the Absolute value of the difference generated by the Subtractor. While the state metric values are used in the ACS recursively, the reliability and decision values are passed to survivor and updating processes.

B. SMU

Fig. 4 (a) shows the SMU architecture used to find the survivor state where all the survivor paths merge, as shown in Fig. 1. Process Elements (PE), shown in Fig. 4 (b), decide the survivor path for each state based on the decision bits (d) provided by the TMU. The SMU shown in Fig. 4 (a) has a similar structure to the SMU structure introduced in [7]. However, our proposed SMU is based on TBA and does not have registers to store the decoded bits while the SMU in [7] is based on REA. In our SMU, the inputs to all PEs at stage k are initialized to '1' to make all states into survivor states. During the survivor process, one of the PE outputs is set to

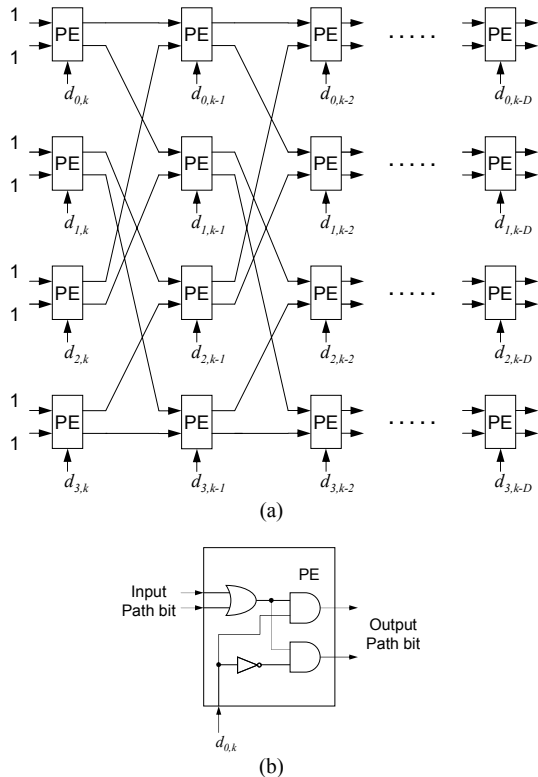


Figure 4. Schematic diagram of (a) SMU for constraint length $K=3$ and (b) PE.

'1' and the other to '0' based on the decision bit (d) and the input path bits as shown in Fig. 4 (b). When both inputs are '0', then both outputs become '0' without depending on the decision bit. Therefore, after a number of stages (i.e., at stage $k-D$) only one output remains as being '1', representing the decoded bit and the final merged survivor state. With our SMU, this survivor process is performed in one clock cycle. Therefore, increasing the decoding depth (D) will increase the critical path delay of the SMU. However, from power consumption point of view, our SMU can save more power than the SMU based on REA.

C. UPU

The UPU module compares the bits generated from the survivor and competing paths to update the reliability values provided by the TMU. Fig. 5 shows the structure of the UPU. While the inputs to all PEs at stage k are initialized to '1' in the SMU in Fig. 4 (a), the inputs of only one PE are initialized to '1' in the UPU. This PE is determined by the final output of the SMU. Fig. 5 shows an example in which the 2nd state is the starting state of the updating process. Then, similar to the SMU described earlier, the UPU provides the decoded bits for the competing paths. These bits can be generated using the outputs of the PEs, which are determined by the decision bits. During the updating process, one output among eight outputs of the PEs always has a value of '1' and all others are '0'. This output indicates the competing path and can be used to identify the competing path bit, c_k , to compare with the decoded bit, b_k , generated by the SMU. In [4], for the updating process, the state information is provided by the SMU to generate the competing path bits. However, in our UPU architecture, all outputs of the PEs are passed to the competing path bit generator (CBG) that outputs the competing path bit (c_k). This bit, c_k , is compared with the decoded bit, b_k , using a XOR. If the output of XOR is '1', the updating process is

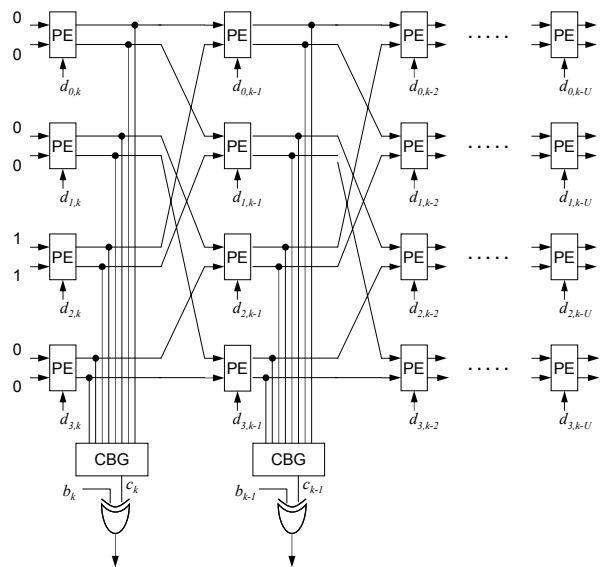


Figure 5. UPU architecture for constraint length $K=3$.

performed following the updating rule as described in section II.

IV. SIMULATION RESULTS

A turbo decoder based on the two-step SOVA algorithm has been simulated with fixed point implementation to verify its functionality. In this simulation, 1/3 code rate, additive white Gaussian noise (AWGN) channel model, and binary phase shift keying (BPSK) were assumed. A block size of 1024 interleaver and deinterleaver is used in the simulation. The input symbols for systematic and parity bits are represented with 16 quantization levels, which can produce BER performance without significant degradation when compared with the performance of real value implementations. On the other hand, in the update process, 32 quantization levels are used for LLR values. In [10], where Log-MAP algorithm is used to implement a turbo decoder, 64 quantization levels are suggested for LLR values. However, for our simulations 32 quantization levels were sufficient to represent the LLR values. These quantization levels are one of the important factors in order to achieve an optimized hardware implementation in terms of area usage and power consumption.

Following the above conditions, we have simulated our two-step SOVA turbo decoder for its BER performance with constraint lengths $K=3, 4,$ and 5 . Fig. 6 (a), (b), and (c) illustrate the BER performance after 1, 2, 4, and 8 iterations. In these simulations, a scale factor for extrinsic information was not considered. The simulation results show the typical characteristics of a turbo decoder. The BER performance in Fig. 6 is dramatically improved after 4 iterations, after which there is no significant improvement in BER.

We have also implemented our two-step SOVA turbo decoder in Verilog HDL in order to evaluate its power consumption and area usage compared to a conventional two-step SOVA turbo decoder. These decoders (our and conventional) were synthesized with Synopsys DesignCompiler™ using the UMC 0.18um standard cell library and simulated at gate level with a clock frequency of 50MHz. Their power consumption was evaluated with Synopsys PowerCompiler™. There is a strong relationship between the survivor and update length parameters and BER performance [11] as well as power consumption and area usage. In our implementations, the survivor and update lengths were fixed to seven times of the constraint length K .

Table I summarizes the implementation results for area usage and power consumption for the two decoders for constraint length $K=5$. As can be clearly seen, our SMU provides 69% saving in area usage and 30% saving in power consumption compared to the conventional SMU. Since the main difference between the two decoders is in their SMUs, the area and power results for their TMU and UPU modules are similar. This results in overall reduction of 46% for area and 15% for power.

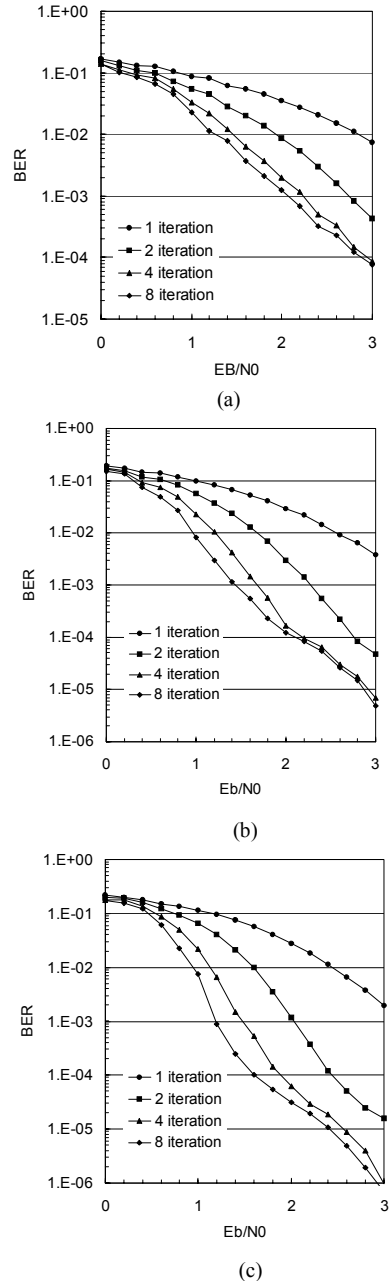


Figure 6. BER performance results for constraint length (a) $K=3$, (b) 4, and (c) 5.

V. CONCLUSIONS

The use of turbo codes in wireless communication systems has been increasing since their invention in the early 1990s. This requires high performance decoder architectures in terms of speed, power, and area metrics. In this paper, a low power and area efficient SISO turbo decoder architecture based on two-step SOVA has been introduced. The power and area savings were mainly achieved by optimizing the survivor memory unit of the two-step SOVA decoder. Our results show that, for constraint length $K=5$, our decoder achieves 46% area and 15% power savings compared to a conventional two-step

SOVA turbo decoder.

Table I. Area and power results for constraint length K=5

Modules	Area (mm ²)			Power (mW)		
	Conv.	Our	Saving (%)	Conv.	Our	Saving (%)
TMU	0.103	0.101	2	6.9	6.6	4
SMU	0.736	0.227	69	20.1	14.1	30
UPU	0.252	0.253	0	12.4	12.8	-3
Total	1.09	0.58	46	39.4	33.5	15

REFERENCES

[1] C. Berrou, A. Gliavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes (1)," in *Proc. IEEE Int. Conf. Communications*, pp. 1064-1070, 1993.

[2] C. Berrou, "The Ten-Year-Old Turbo Codes are Entering into Service," *IEEE Commun. Mag.* vol. 41, no. 8, pp.110-116, Aug. 2003.

[3] J. Hagenauer and P. Höher, "A Viterbi Algorithm with Soft Outputs and It's Application," in *Proc. IEEE Global Commun. Conf.*, pp. 47.1.1-47.1.7, Nov. 1989.

[4] H. Dawid, and H. Meyer, "Real-time algorithms and VLSI architectures for soft output MAP convolutional decoding", *IEEE Int. Sym. PIMRC*, Vol. 1, pp.193-197, Sep. 1995.

[5] Engling Yeo, Stephanie A Augsburg, W. Rhett Davis, Borivoje Nikolic, "A 500-Mb/s Soft-Out Viterbi Decoder," in *IEEE J. Solid State Circuits*, Vol. 38, pp.1234-1241, July 2003

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

[7] Olaf J. Joeressen, Martin Vaupel, and Heinrich Meyr, "High-Speed VLSI Architectures for Soft-Output Viterbi Decoding," *Proc. IEEE Int. Conf. Application Specific Array Processors*, pp. 373-384, Aug. 1992.

[8] Z. Wang, and Keshab K. Parhi, "High Performance, High Throughput Turbo/SOVA Decoder Design," in *IEEE Trans. on Comm.*, Vol. 51, No. 4, pp. 570-579, 2003.

[9] C. Berrou, P. Adde, E. Angui, and S. Faudeil, "A low complexity soft-output Viterbi algorithm," in *Proc. IEEE Int. Conf. Communications*, pp. 737-740, 1993.

[10] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal decoding algorithm", *Proc. IEEE Int. Conf. Commun.* pp. 1009-1013, 1995.

[11] Olaf J. Joeressen, Martin Vaupel, and Heinrich Meyr, "Soft-Output Viterbi Decoding: VLSI Implementation Issues," *IEEE Vehicular Technology Conf.*, pp. 941-944, 1993.