ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

Power Efficient Survivor Memory Architecture for Viterbi Decoder

[1] G.Sankar Babu, [2]Dr.M.Anto Bennet, [3]R.Kaushik Krishna, [4] B.S.Jaya Vignesh, [5]B.Ashwin

[1] Assistant Professor, Dept. of ECE, VELTECH, Avadi, Chennai, Tamilnadu, India

[2] Professor, Dept. of ECE, VELTECH, Avadi, Chennai, Tamilnadu, India

[3,4,5] UG Students, Dept. of ECE, VELTECH, Avadi, Chennai, Tamilnadu, India

ABSTRACT: Viterbi decoder is a common module in communication system in which power and decoding latency are constraint. Register exchange (RE) architecture has the lowest decoding latency L. However, it is not suitable for communication system because of its high power consumption. In this paper, we propose a new SMU architecture which combines the concept of the trace-forward and trace-back. The decoding latency of the proposed SMU algorithm is only L+M. Besides, we present a power efficient architecture for the proposed SMU algorithm.The power consumption of the proposed architecture is slightly higher than the 3-pointer even TB architecture.

I. INTRODUCTION

Convolution code (CC) is an essential forward error correcting code (FEC) for many wireless communication systems, such as WiMAX and 3G systems. The Viterbi algorithm (VA) is known as an optimal decoding approach for convolution code.For the application of the high data rate wireless communication systems, it is important to reduce decoding latency and power consumption in Viterbi decoder.

A Viterbi decoder is composed of three main blocks:

Branch Metric Unit (BMU): calculates the distancebetween received codewords and referenced codewords;

Add-Compare-Select Unit (ACSU): selects a best trellispath based on current BM and previously state metrics;

Survivor Memory Unit (SMU): generates the decodedbits by the best state sequence in the trellis.

In general, ACSU and SMU are 2 major power consuming modules in the whole Viterbi decoder. Besides, the decoding latency is determined by the SUM architecture used in the Viterbi decoder. There are 2 common SMU architectures, Register Exchange(RE)andTrace-back(TB). Both 2 architectures utilize the merge phenomenon of survivor paths after decoding length L. In this paper, we consider convolutional code that has $N = 2^M$ states in the trellis, where $M$is the number of the registers in the CC encoder.

RE has the lowest decoding latency and simple control circuit. RE is implemented by the connection of multiplexers and registers according to the trellis diagram, and its memoryrequirement is NL bits registers. As decoding, all NL bits are read and written, and it requires high memory access bandwidth and consumes high power. On the other hand, the power consumption of the TB architecture is much more efficient than RE.

In TB method, only N decision bits are written in each cycle, and RAM can be utilized here. Therefore, TB is more suitable for the application of communication system. However, the drawback of the TB method is its long decoding latency. There are two solutions for this drawback. One is to use more memory banks and access pointers,

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

such as 3-pointer TB architecture. Another is to use high radix ACSU, which will increases the area and power consumption of the ACSU. Recently, a pre-traceback SUM architecture has been proposed in, and it combines the TB method and trace- forward (TF) technique. The TF unit can reduce the decoding latency and the memory read operation.

In this paper, we propose a SMU algorithm named *state exchange* (SE). The SE method uses the TF units as the key decoding module rather than an assistant module in pretraceback method. The power consumption of the SE architecture is slightly higher the TB method, and the decoding latency is only $L+M$. Hence, the Viterbi decoder does not need to use high radix ACSU as using the SE architecture.

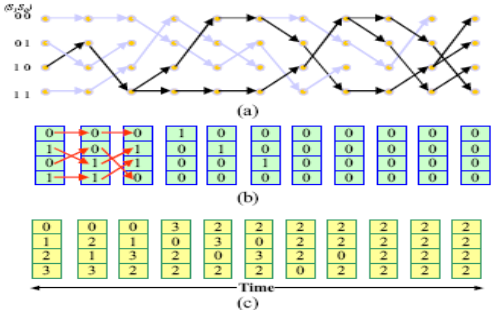
The remainder of the paper is organized as: In section II, we give some background knowledge. Section III describes the proposed SE algorithm. Section IV presents a power efficient architecture for the SE algorithm. Section V gives a comparison and Section VI concludes simulation results of the paper.

II. TRACE-FORWARD TECHNIQUE

In conventional TB method, it is necessary to get the initial start state for the decoding operation. In the TF unit is utilized to replace the backward operation. Hence, the TF unit can reduce the latency of the SMU.

A. Basic concept of the Trace-forward Technique

The concepts of the TF algorithm and the RE are similar. The difference between RE and TF lies in the register contents: the register contents in RE are initialized with 0 or 1 and then interchanged, while the register contents in the TF unit are initialized with state information. A simple example with L=10, M=2, and$N$=4. is illustrated in Fig. 1.



Fig,1. Trellis stage

In Fig. 1 (a), the trellis states are denoted as (S1, S0) = 00, 01, 10, 11. Each branch is a survivor path transition determined by the decision bits from the ACSU. The black branch is the final converged path, which can be utilized for decoding. Fig. 1 (b) shows the operation of the register in the RE scheme, and Fig. 1 (c) is the TF technique. Both 2 schemes exchange the contents of the register like the survivor path transition. In this example, the initial decoding bit is 0, and the initial start state is state 2. After 10 iteration, the register contents of 2 schemes all converges.

B. The Architecture of the Trace-forward Unit

I The TF unit is utilized to get an initial decoding state. Hence, the TF unit only needs N register with M bits. Fig.2 shows a TF register named ^{n}
updated as follows:

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

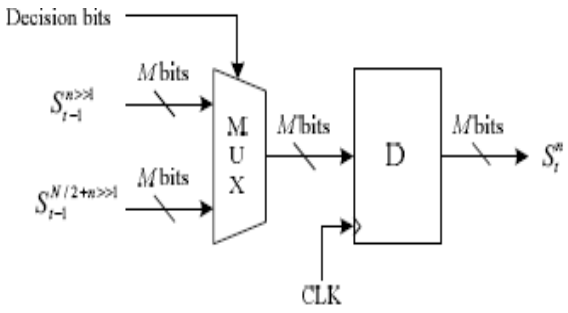


Fig2. Trace forward architecture

III. PROPOSED STATE - EXCHANGE ALGORITHM

The proposed state exchange method utilizes the TF unit, which interchange the register contents according to the decision bits. Hence, the SE algorithm is similar to the conventional RE method. The basic concept and the operation of the SE algorithm are discussed in the following.

A. Basic concept

The SE algorithm adopts a new concept when retrieving the decoded sequence. Before presenting the SE algorithm, there are two points of view should be clarified:

In RE scheme, there are *L* stages of registers, each stage is composed of *N* (state number) one-bit registers. The registers in the first stage are fed by 0 or 1 according to their states, and the contents of register are interchange between each stage according to the decision bits generated by the ACSU. After *L* stages, the contents of the last stage's registers are expected to be the same due to the merge phenomenon of the survivor paths, and the decoded bit can be derived from the register contents. While in TF unit, there is only one stage of registers, and each register is initialized to their state number. After *L* iterations, each register is expected to have the same state number, and the state is the converged state before *L* iterations, as shown in Fig. 1.

In the converged state of the TF unit is considered as the starting state of the decoding operation. We, however, find out that the converged state contains other information: The converged state itself is not only the starting state of the decode operation but also the decoded bits for *M* iterations. It means that the trellis state is the last *M* bits fed into the encoder while encoding, which can be easily shown in Fig.2. In this figure, we take a convolutional encoder with four shift register as example. If the trellis state (S3, S2, S1, S0) = (1, 1, 0, 1) at this time, the last four bits from the input must also be 1, 1, 0, 1. In other words, *M*decoded bits can be derived from the converged state. Hence, we can use several TF units to track several converged states which are spaced *M* time instant. This is the main concept of the SE algorithm.

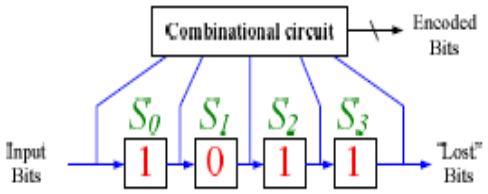


Fig .3. Feed forward convolution encoder

B. The architecture of the SE algorithm

As mentioned above, we can get ***M*** decoded bits from the converged state. The register contents of the TF unit converge After running *L* iteration. There should be *K* TF units (TFU) in our SMU architecture, and *K* is determined by *L* and *M*:

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

k=L/M

For simplicity, we use an example to illustrate the decoding operation of the proposed SE algorithm, as shown in Fig.4. The example is provided with a four state trellis (*N*=4, and *M*=2). Besides, we assume that the survivor paths will converge in *8* iterations. The total number of the TFU is 4

$$\left\lceil \frac{8}{2} \right\rceil = 4 .$$

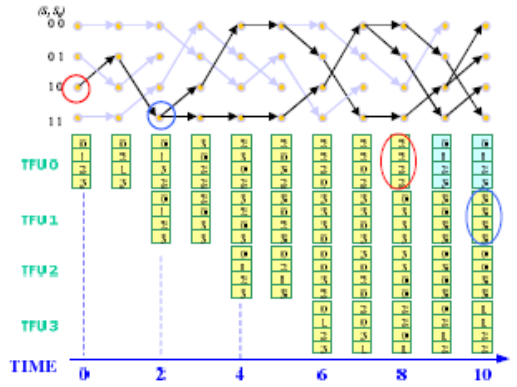


Fig4.decoding operation of SE algorithm

The TFU 0 is utilized to trace the state at time instant 0, which is within the red circle. At time instant 8, the contents of the TFU 0 are all converged to state 2. Similarly, the other three trace-forward units are initialized and start transition at time 2, 4, and 6, respectively. As mentioned above, the converged state represents the *M* decoded bits *L* clocks before, where *L* represent the time period between the current time and the time when this TFU is initialized. We thus know that the decoded bits are *1* and *0* at time instant -2 and -1 respectively. Similarly, the contents of the TFU 1 converge to 3 at time 10. That means the converged state at time instant 2 is 3, and the decoded bits are 1 and 1 at time instant 0 and 1 respectively. At time instant 8, the contents of the TFU 0 has been used, and the TFU 0 can be initialized again for further decoding process. By this method, four TFUs are used circularly, and the decoding process can be continuous without any usage of SRAM cells. It should be noted that there is no input data at time instant -2 and -1. Hence, the proposed SMU starts operating at time instant *M*in practical application.

However, the architecture shown in Fig.3 cannot be better than the RE method. From Fig. 3, we can easily see that four TFUs are running concurrently at any time instant while decoding. Therefore, the architecture above brings huge power consumption due to the interchanging of contents in every register, just like the RE method. In section IV, we will give a power efficient architecture for the proposed SE algorithm.

IV. POWER EFFICIENT ARCHITECTURE FOR THE SE ALGORITHM

The main concept and a simple decoding example of the SE algorithm have been introduced above. We develop a new architecture which consumes much less power than the architecture shown in Fig. 3.

This power efficient architecture utilizes the fact that there is a link between 2 successive survivor states. Because the survivor states are stored in the TFUs, it is possible to get a converged state by tracing the survivor states. Hence, to derive a converged state, the TFUs only need to run *M* iterations rather than *L* iterations. We also give an example as shown in Fig. 5:

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

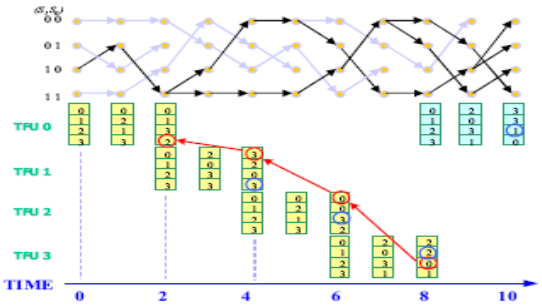


Fig. 5 The power efficient decoding operation of the proposed SE algorithm

For convenience, we use the same example as Fig.4. The major difference is that each TFU only works for 2 iterations. There is only one TFU working at any time instant. In Fig. 4, the converged state at time instant 0 is state 2. Now the converged state can be derived by tracing the survivor state in other 3 TFUs. For example, we select the 2nd register in the TFU 3 and check the register content. Then we trace back to the 0th register in the TFU 2. Similarly, we trace back to the 0th register in the TFU 2. Finally, we can get the content of the 4th register in the TFU 0, which consists with the converged state in Fig. 4. It should be noted that the contents in the TFU 0 has been utilized at time instant 8. Hence, the TFU 0 can be initialized and restarts working after time 8. Using the same method, we can get the converged state at time instant *2* by the backward tracking process through TFU 0, TFU 3, TFU 2, and then TFU 1. Adopting this method circularly, the decoding task can be continuously processed with only one TFU working at any time. We can turn off the TFUs that are not working. Therefore, the power consumption can be greatly reduced. The final hardware architecture is shown in Fig.6:

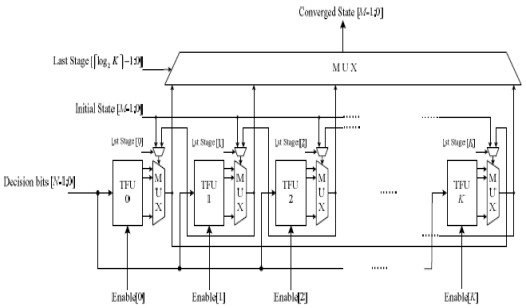


Fig.6.Hardward architecture

It should be noted that the decoding depth L must be the multiple of M. The signal description is shown in Table 1:

Table 1 Signal Description

Signal Name	Description
Enable[i]	1: make the TFU i working; 0: turn off
1st Stage[i]	1: the TFU i is the 1st stage in the tracing back operation; 0: otherwise.
Initial State	Define the selected register number in the 1st stage TFU.
Last Stage	Define the number of the TFU that is the last stage of the trace back

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

If the decoding depth *L* is large enough, the signal "Initial State" can be a random number.

V. COMPARISON

	Voltage (V	Current (m	Power (m
Vccint	1.2		
Dynamic		72.80	87.36
Quiescent		0.00	0.00
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		15.00	37.50
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
Total Pow			124.86
Startup Curre		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig:7.Power consumption of RE Method

	Voltage (V	Current (m	Power (m
Vccint	1.2		
Dynamic		6.89	8.27
Quiescent		0.00	0.00
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		15.00	37.50
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
Total Pow			45.77
Startup Curre		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig:8.Power consumption of REMethod

By using the architecture shown in Fig. 6, we can get a SMU with the decoding latency near the RE method with much less power consumption. In this section, we use a CC encoder specification in a practical system. The constraint length is 7 (N=64, and M=6), and the generator polynomial is (171, 133). Besides, we implement the Viterbi decoder with L=36.

Fig 7 and 8 shows the comparison of power consumptio between the RE method, the proposed SE method. We use the 3-pointer even architecture for the TB method. We use VHDL to implement three architectures, and From the simulation, it is clear that the power consumption is slight higher than the 3-pointer even architecture. This feature makes the proposed SE architecture being able to be implemented in communication system.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

VI. SIMULATION RESULTS

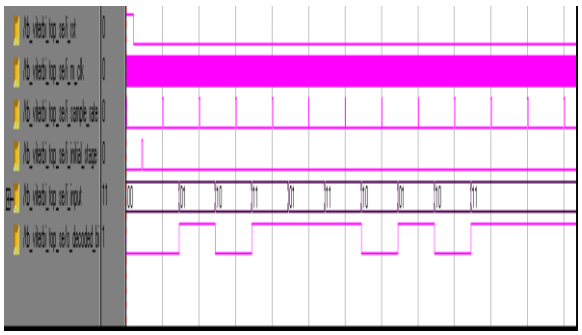


Fig 9.Summarized Viterbi decoder output

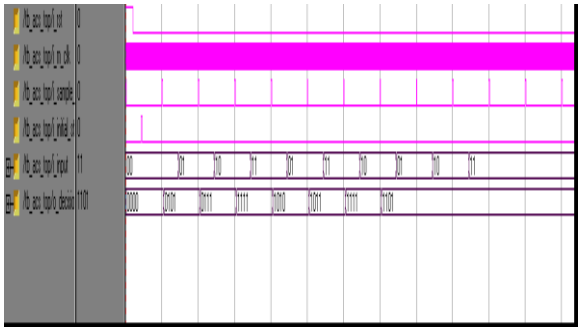


Fig: 10. ACS unit output

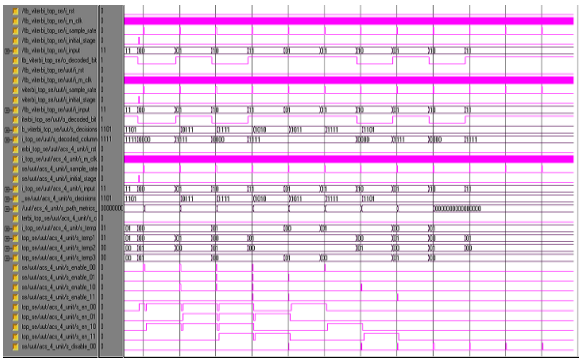


Fig: 11. Output of all units in Viterbi

REFERENCES

1. J. Viterbi, "Convolutional codes and their performance in communication systems," IEEE Trans. Commun., Vol. COM-19, no. 10,pp. 751-771, Oct. 1971.
2. G. Feygin and P. G. Gulak, "Architecural tradeoffs for survivor sequence memory management in Viterbi decoders," IEEE Trans. Commun., vol.41, no.3, pp. 425-429, Mar. 1993.
3. P. J. Blck and T. H. Meng, "A 140-mb/s 32-state, radix-4 Viterbi decoder," IEEE J. Solid-State Circuits, vol.27, no.6, pp. 1877-1885, Dec. 1992.
4. Y. Gang, A. T. Erdogan, and T. Arslan, "An efficient pre-traceback architecture for the Viterbi decoder targeting wireless communication applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vo1. 52, no. 6, pp.1148–1156, Jun, 2005.