A Viterbi decoder with low-power trace-back memory structure for wireless pervasive communications.

Pasin Israsena[1]
Izzet Kale[2,3]

[1] Thailand IC Design Incubator (TIDI), National Electronics and Computer Technology Center
[2] School of Informatics, University of Westminster
[3] Applied DSP and VLSI Research Centre, Eastern Mediterranean University

A Viterbi Decoder with Low-Power Trace-Back Memory Structure for Wireless Pervasive Communications

Pasin Israsena [#]
#Thailand IC Design Incubator (TIDI)
National Electronics and Computer Technology Center
112 Thailand Science Park, Paholyothin Rd
Pathumtani, Thailand 12120
Email: pasin.israsena@nectec.or.th

Izzet Kale [*+]
*Applied DSP and VLSI Research Group
Department of Electronic Systems
University of Westminster, London, United Kingdom
+Applied DSP and VLSI Research Centre
Eastern Mediterranean University
Gazimagusa, Mersin 10, Turkey
Email: kalei@westminster.ac.uk

Abstract— **This paper presents a new trace-back memory structure for Viterbi Decoder that reduces power consumption by 63% compares to conventional RAM based design. Instead of the intensive read and write operations as required in RAM based designs, the new memory is based on an array of registers connected with trace-back signals that decode the output bits on the fly. The structure is used together with appropriate clock and power-aware control signals. Based on a 0.35 µm CMOS implementation the trace-back back memory consumes energy of 182 pJ.**

Index Terms—**Viterbi decoder, trace back memory, low-power, channel coding, convolutional code**

I. INTRODUCTION

Convolution coding is widely used in modern digital communication systems such as mobile or satellite communications to achieve low-error rate data transmission. The Viterbi algorithm [1], in particular, is known to be an efficient method for the realisation of maximum- likelihood (ML) decoding of the convolutional codes. Today Viterbi Decoder is widely used in established systems such as GSM mobile or IEEE 802.11a wireless LAN standard. With emerging applications such as Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB) or wearable personal entertainment devices, wireless communication is also increasingly becoming more pervasive. These applications require devices with ultra low power consumption. Already it has been shown that the Viterbi decoder can account for more than one third of power consumption during baseband processing in second-generation cellular telephones [2]. Power consumption is therefore the critical design criteria to be tackled.

An example in Figure 1 shows a 4-state (K=3) convolutional system with the coding rate (number of input bits/output bits) R of ½. The corresponding Trellis diagram is shown in Figure 2. The states are presented in Y-axis, with timing in X. For each branch between each pair of states, the output values for that transition is stated. A convolutional code is often represented as *(n,k,K)code* where n is the number of input bits, k the number of output bits, and K is the encode constraint length. To decoder data, a conventional Viterbi decoder consists of 3 major parts that are:

- Branch metric computation unit (BMCU)
- Add-compare-select unit (ACSU) containing ACS cells(s) and PMU
- Survivor memory unit (SMU)

For an encoder with constraint length L =K, the BMU works to find the likelihood of each of the $2^{(K-1)}$ states of the decoder transferring to a next state under particular set of input symbols. The ACSU compares the results to find maximum likelihood for each state, updates its path metrics, and generates a decision bit that uniquely identify the previous or surviving states. These decision bits are stored in SMU and used to reconstruct the most likely state sequence.

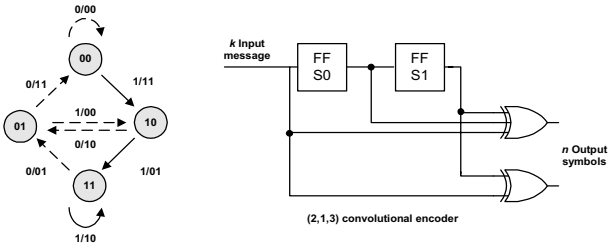


Figure 1. State diagram

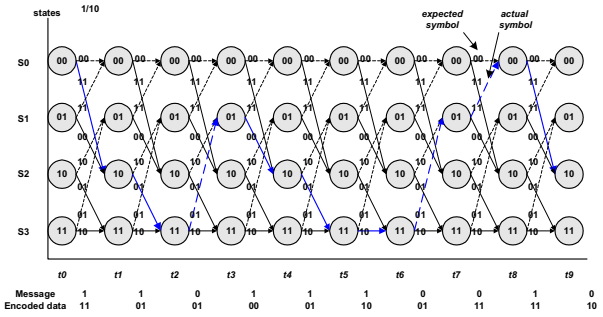


Figure 2. Trellis diagram of (2,1,3) convolutional code

Viterbi decoder has been a subject of ongoing research, the most recent ones such that [3-5] concern high-speed implementations. Approaches for power reduction have also proposed [6-10]. In this paper we introduce low-power design techniques targeting specifically the key power hungry block within the Viterbi decoder, which is the SMU. An SMU usually employs either register exchange or trace back techniques. In register exchange designs [11] the decoded bits for each state under maximum likelihood path are stored directly onto the state registers. The approach consumes significant power and is suitable only for designs with short decoder length, as the decoded data stored, which needs constant read/write updates, increases with the length of the decoding process. For systems that require longer decoding length for better bit error rate, such as those in GSM or CDMA, the trace-back technique [12] is usually employed. Using trace-back algorithm, only information which identifies previous unique states is kept in the memory, meaning only one bit/state/stage is required. After a certain decoder length the information is read out and decoded to find the most likely incoming bit sequences received during that period. The trace-back SMU is conventionally realised using RAM, and as a result it has been shown to contribute more than half of the power consumption in a conventional Viterbi decoder due to the expensive memory accesses [8].

This paper proposes a new memory structure where decoded bits are presented without having to be read out. The new memory is based on an array of registers connected with trace-back signals that decode the output bits on-the-fly. The structure is used together with appropriate clock and power-aware control signals. The detail implementation is given in the next section.

II. PROPOSED TRACE-BACK MEMORY ARCHITECTURES

The new memory structure is based on connections of new cell elements, each has a structure as shown in the Figure 3. The cell consists of a register with additional logics to allow the stored trace-back information to be read. The logics are also connected to other cells such that, based on the bit information stored, it can uniquely identify the previous or surviving state so that the decoded bit can be found. An example of the memory structure for a 64-state Viterbi decoder with a decoding length of 35 is shown in Figure 4. Potentially power consumption is minimised as, once stored in the register bits, the result are already presented without have to be read out again as in the case of using RAM. The memory structure is also designed such that only once the winner is decided upon that the signals are to be propagated through the array to decode the output bits. Apart from that period, no switching of decoding logics occurs. In terms of writing the data onto the registers, two approaches are considered. The first approach, termed systolic (figure 5), is when the data propagates through the stage (or column) resisters under the same clock rate. For the second approach, termed parallel (Figure 6), data is connected to all the column registers in parallel, each clocked at clk/35 rate and appropriately skewed. The results are discussed next.

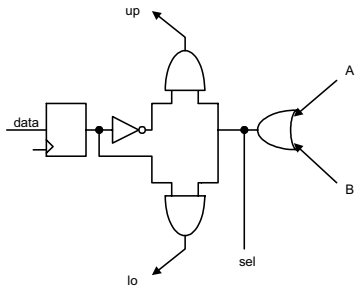


Figure 3. cell structure

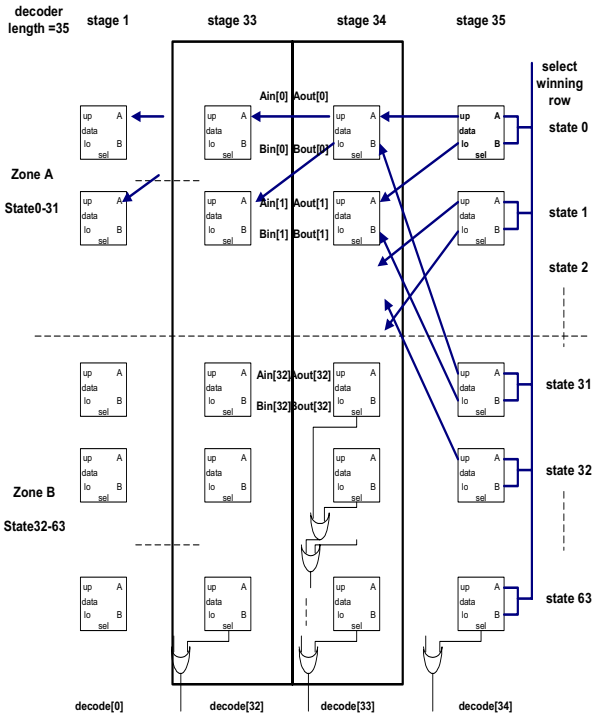


Figure 4. New Trace-bank memory connection

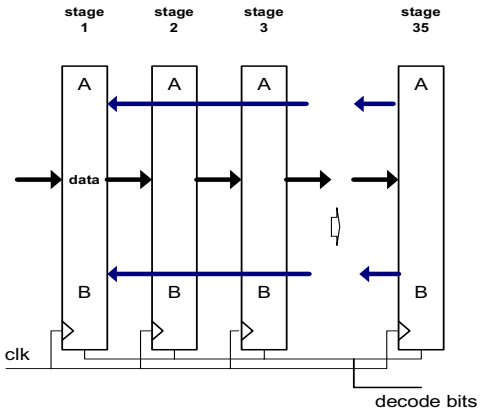


Figure 5. Systolic architecture

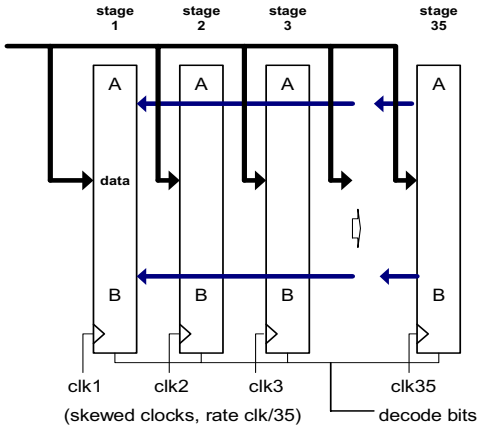


Figure 6. Parallel architecture

III. RESULTS AND DISCUSSIONS

To test the effectiveness of the new memory structure, a Viterbi decoder that follows DAB specifications is constructed. The DAB specifications have a constraint length *L* of 7 (hence 64 states) rate *R* of ¼ , with the encoder as shown in Figure 7.

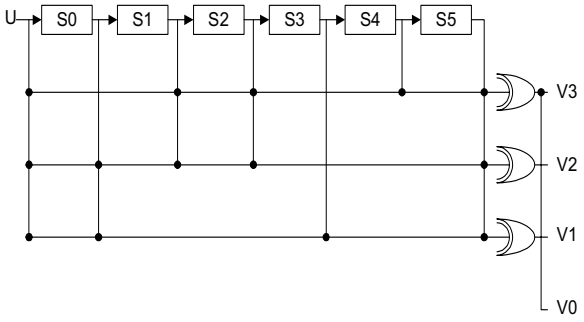


Figure 7. DAB encoder

Although the memories considered can be easily adapted for both single and multiple ACU cells approaches, for simplicity the Viterbi structure implemented here has a single state ACS unit, requiring 64 clocks per one stage operation. Also, as it is generally considered that the decoder length should be about five times the constraint length, the trace-back has memory depth of 35. The block diagram of the decoder is as shown in Figure 8.

The Viterbi decoder based on the three SMU architectures that are RAM based, the proposed systolic and parallel forms are implemented using Verilog HDL and are functionally verified both by simulation on Mentor Graphic's ModelSim and for Xilinx FPGA implementation. Examples of the simulation result showing the decoders correctly decoding the sample data (hex 713922f2b) are shown in Figure 9. The designs are synthesised for ASIC implementation of SMU using Mentor Graphic's LeonardoSpectrum using 0.35µm CMOS technology. Power dissipation is estimated based on switching activity observations and information provided by [13]. The results are given in Tables 1 and 2.

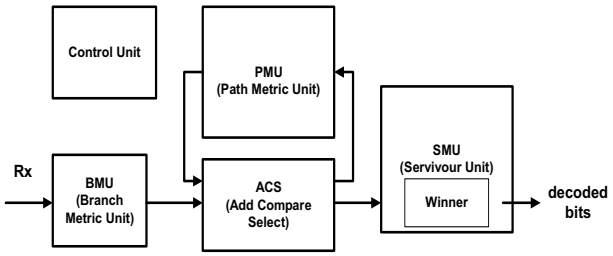
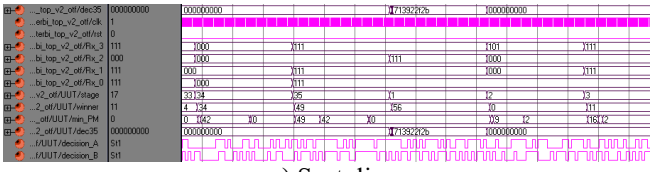
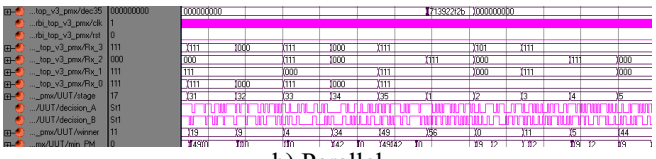


Figure 8. Block diagram of Decoder hardware



a) Systolic



b) Parallel

Figure 9. Simulation results

TABLE I. VITERBI DECODER USING FPGA

FPGA XV1000E	Slice No.	Utilize	Delay ns	Max rate MHz
RAM	305	2%	12.9	77.5
Systolic	4526	36%	8.268	120.9
Parallel	4580	37%	16.598	60.25

TABLE II. NEW TRACE-BACK MEMORY ASIC IMPLEMENTATION

ASIC 0.35µm	**Systolic**	**Parallel**
Total Energy/Sample (pJ)	539	182
- Registers	425	23.6
- Conbinational	15.7	30.5
- Interconnects	31.4	61
- Clock	67	67
% Energy/Sample compare to RAM	108	36.4
Area (mm^2)	1.19	1.20
Delay (ns)	20.37	20.45

Table 1 shows results of FPGA implementations of the Viterbi decoder employing different types of memory. It can be seen that the RAM based design uses significantly less resource. This is due to the fact that FPGA is a RAM based technology and so the resource required for realising arrays of FFs as required by systolic and parallel structures is considerable. This is particularly true for heavily routed structure such as the parallel design, where its delay is also significantly increased. FPGA is however, inherently a power-hungry technology needing large static current, and so the power aspect is not considered and the implementations are only for verification purposes. For low-power ASIC implementation, the SMU based on the systolic and parallel designs are implemented and compared to the 2Mbit macro block RAM given in [13] that can be employed for the same purpose. It can be seen that the parallel design consumes much less power (energy). This is due mainly to the reduction in power due to low switching FFs given slower rate of clocking for each column registers. This is compared to the FFs in the systolic design that switch, depending on the particular stage registers, at a rate much closer to the decode rate. The reduction in register power is traded-off slightly by the increase in power due to increase in routing resulting in larger power for interconnects and gate capacitances. Compare to RAM, the power consumption of the new trace-back memory based on the parallel design is only 36.4%. The result also compares favorably with previously reported design such as [14] which uses similar CMOS technology. Given that SMU contributes more than 50% of the total power consumed by a Viterbi decoder, by using the new memory structure the reduction in power in total is potentially more than 30%. In the cases where an increase in the area is not priority, using register based design is also more appropriate for soft IP approach, as the code can be portable more easily compared to designs using RAM where a macro RAM block usually needs to be provided by the vendor. In terms of speed, Table 2 also provides the delays of both new memory designs. Although the values are adequate for most applications, implementation using a deeper submicron technology will also allow the memory to be used in more demanding, ultra high-speed applications. A modification to the structure for high-speed applications is also potentially possible and is the subject of our ongoing work.

IV. CONCLUSIONS

A new trace-back memory structure for Viterbi Decoder that reduces power consumption by 63% compares to the conventional RAM based design is proposed. The new memory is based on an array of registers connected with trace-back signals that decode the output bits on thefly. The structure is used together with appropriate clock gating and power-aware control signals. Based on 0.35 µm COMS implementation the trace-back back memory consumes and energy of 182 nJ.

REFERENCES

[1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, 1967.

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

[3] K.K Parhi, "An improved pipelined MSB-first add-compare select unit structure for Viterbi decoders", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 51, pp. 504 – 511, March 2004

[4] Liu Xun and M.C Papaefthymiou, "Design of a 20-mb/s 256-state Viterbi decoder," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, Issue 6, pp. 965 – 975, Dec. 2003

[5] Liu Xun and M.C Papaefthymiou, "Design of a high-throughput low-power IS95 Viterbi decoder", *Proc. 39th Design Automation Conference*, pp. 263-268, June 2002.

[6] D. Garrett and M. Stan, "Low power architecture of the soft-output Viterbi algorithm," *Proc. ISLEPD '98*, pp. 262-267, 1998.

[7] I. Kang and A.N. Willson Jr, "Low-power Viterbi decoder for CDMA mobile terminals", *IEEE-Journal of Solid-State Circuits.* vol.33, no.3, pp.473-82, March 1998

[8] D. Oh and S. Hwang, "Design of a Viterbi decoder with low power using minimum transition traceback scheme," *Electronic Letters*, vol.32, no.22, pp. 2198-2199, Oct. 1996.

[9] K. Seki; S. Kubota, M. Mizoguchi, and S. Kato, "Very low power consumption Viterbi decoder LSIC employing the SST (scarce state transition) scheme for multimedia mobile communications," *Electronics Letters,* vol.30, no.8, pp.637-639, April 1994.

[10] F. Ghanipour and A.R. Nabavi, "Design of a low-power Viterbi decoder for wireless communications," *Proc. IEEE ICECS 2003*, vol. 1, pp. 304-307, Dec. 2003

[11] Gennady Feygin and P.G. Gulak. "Architectural Tradeoff for Survivor Sequence Memory in Viterbi Decoder." *IEEE Transactions on Communications*, Vol. 41, pp. 425-429, 1998.

[12] R. Cypher and C. B. Shung, "Generalized trace-back techniques for survivor memory management in the Viterbi algorithm," *IEEE J. VLSI Signal Processing*, vol. 5, pp. 85–94, Jan. 1993.

[13] AustriaMicroSystems 0.35um CMOS Digital Standard Cell Databook, 2003

[14] Chien-Ching Lin, Chia-Cho Wu, and Chen-Yi, "A Low Power and High Speed Viterbi Decoder Chip for WLAN Applications", 2003