International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 8 (September 2013), PP. 26-32

Design and Implementation of High Speed Low Power Viterbi Decoder

BaburaoKodavati[1], Subhrajeeth Pradhan[2], G.Naveen Kumar[3], Srikanth patnaiak[4]
[1, 3, 4]Asst.Prof,Giet,Gunupur,Rayagada,Odisha-765022.
[2]Hod Of Ece Department,Giet, Gunupur, Rayagada, Odisha-765022.

Abstract:- Convolution encoding with Viterbi decoding is a powerful method for error checking. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels. The Viterbi algorithm, which is the most extensively employed decoding algorithm for convolution codes. In this paper, we present an implementation of Viterbi Decoder for code rate of ½ and for constraint length of 9 which is employed in present technologies. The Viterbi algorithm is commonly used in a wide range of communications and data storage applications.Thispaper discusses about architecture for a Viterbi Decoder using VLSI design techniques at circuit level. The Viterbi decoder comprises of BMU, PMU and a SMU. Communication within the decoder blocks is controlled by the Request-Acknowledge handshake pair which signals that data is ready for process. The design of various units of Viterbi Decoder is done by VERILOG HDL language. The simulation results show the decoder output of the encoded bits which are nothing but the bits which we are applied to conventional encoder.

I. INTRODUCTION

Convolution coding has been used in communication systems including deep space communications and wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolution coding is that it can be applied to a continuous data stream as well as to blocks of data. IS-95, a wireless digital cellular standard for CDMA (code division multiple access), employs convolution coding. A third generation wireless cellular standard, under preparation, plans to adopt turbo coding, which stems from convolution coding. The Viterbi decoding algorithm, proposed in 1967 by Viterbi, is a decoding process for convolution codes in memory-less noise [52]. The algorithm can be applied to a host of problems encountered in the design of communication systems [52]. The Viterbi decoding algorithm provides both a maximum-likelihood and a maximum a posterior algorithm. A maximum a posterior algorithm identifies a code word that maximizes the conditional probability of the decoded code word against the received code word, in contrast a maximum likelihood algorithm identifies a code word that maximizes the conditional probability of the received code word against the decoded code word. The two algorithms give the same results when the source information has a uniform distribution.

Traditionally, performance and silicon area are the two most important concerns in VLSI design. Recently, power dissipation has also become an important concern, especially in battery- powered applications, such as cellular phones, pagers and laptop computers. Power dissipation can be classified into two categories, static power dissipation and dynamic power dissipation Typically, static power dissipation is due to various leakage currents, while dynamic power dissipation is a result of charging and discharging the parasitic capacitance of transistors andwires. Since the dynamic power dissipation accounts for about 80 to 90 percent of overall power dissipation in CMOS circuits; numerous techniques have been proposed to reduce dynamic power dissipation. These techniques can be applied at different levels of digital design, such as the algorithmic level, the architectural level, the gate level and, the circuit level.

A viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using Forward error correction based on a Convolution code. The Viterbi algorithm is commonly used in a wide range of communications and data storage applications. It is used for decoding convolution codes, in baseband detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for the Viterbi decoder or Viterbi detector, which is a processor that implements the Viterbi algorithm, depend on the applications where they are used. This results in very wide range of required data throughputs and power or area requirements.

Viterbi detectors are used in cellular telephones with low data rates, of the order below 1Mb/s but with very low energy dissipation requirement. They are used for trellis code demodulation in telephone line modems, where the throughput is in the range of tens of kb/s, with restrictive limits in power dissipation and the area/cost of the chip. On the opposite end, very high speed Viterbi detectors are used in magnetic disk drive read channels, with throughputs over 600Mb/s. But at these high speeds, area and power are still limited.

Convolutional coding has been used in communication systems including deep space communications and wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. IS-95, a wireless digital cellular standard for CDMA (code division multiple access), employs convolutional coding.

II. ARCHITECTURE OF VITERBI DECODER

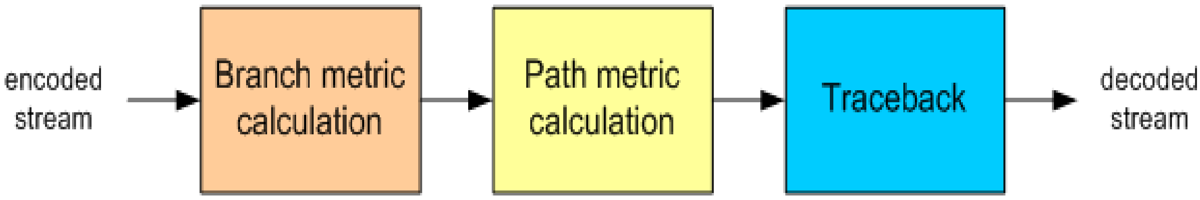


FIG-2.1 Architecture of the VITERBI DECODER

In the viterbi decoding approach the trace-back (TB) and the register-exchange (RE) methods are the two major techniques used for the path history management in the chip designs of Viterbi decoders. The TB method takes up less area but requires much more time as compared to RE method because it needs to search or trace the survivor path back sequentially. But the major disadvantage of the RE approach is that its routing cost is very high especially in the case of long-constraint lengths and it requires much more resources.

2.2 Branch Metric Computation (BMC)

For each state, the Hamming distance between the received bits and the expected bits is calculated. Hamming distance between two symbols of the same length is calculated as the number of bits that are different between them. These branch metric values are passed to Block 2. If soft decision inputs were to be used, branch metric would be calculated as the squared Euclidean distance between the received symbols [21]. The squared Euclidean distance is given as $(a_1\text{-}b_1)^2 + (a_2\text{-}b_2)^2 + (a_3\text{-}b_3)^2$ where a_1, a_2, a_3 and b_1, b_2, b_3 are the three soft decision bits of the received and expected bits respectively.

value	Meaning
000	strongest 0
001	relatively strong 0
010	relatively weak 0
011	weakest 0
100	weakest 1
101	relatively weak 1
110	relatively strong 1
111	strongest 1

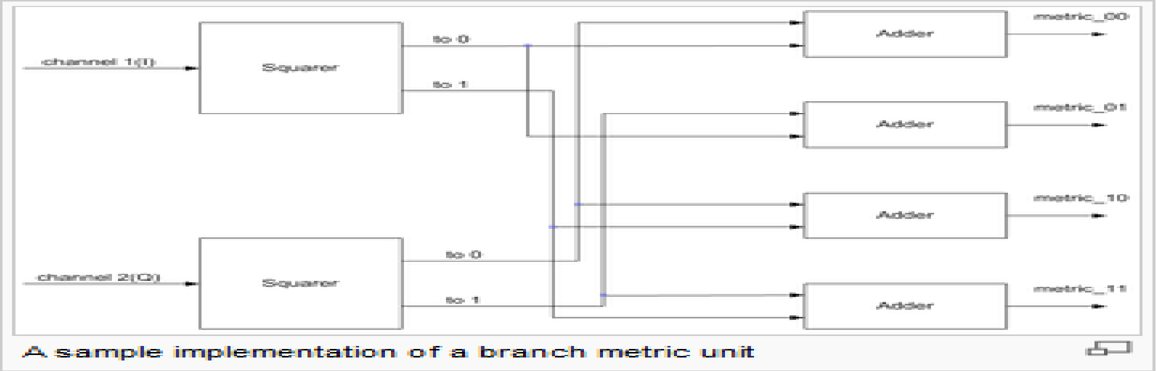


FIG-2.2 A sample implementation of a branch metric unit

2.3 Path Metric Computation and Add-Compare-Select (ACS) Unit

The path metric or error probability for each transition state at a particular time instant is measured as the sum of the path metric for its preceding state and the branch metric between the previous state and the present state. The initial path metric at the first time instant is infinity for all states except state 0. For each state, there are two possible predecessors. The mechanism of calculating the predecessors (and successors) The path metrics from both these predecessors are compared and the one with the smallest path metric is selected. This is the most probable transition that occurred in the original message. In addition, a single bit is also stored for each state which specifies whether the lower or upper predecessor was selected.

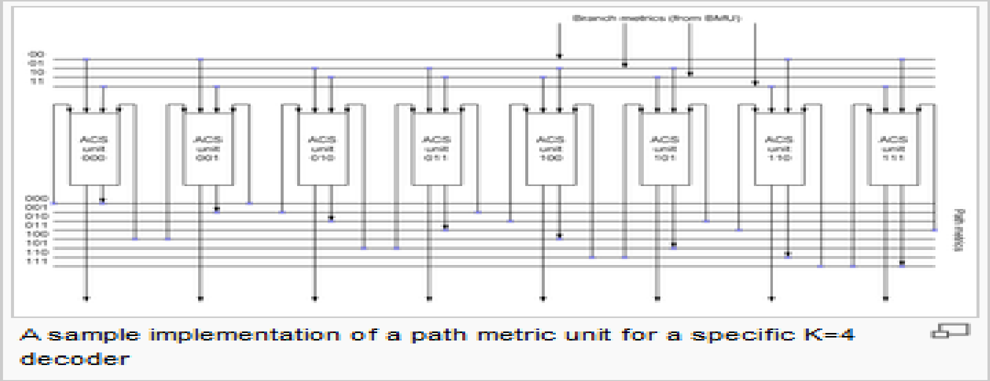


Fig-2.3 A sample implementation of the path metric unit

In cases where both paths result in the same path metric to the state, either the higher or lower state may consistently be chosen as the surviving predecessor. For the purpose of this paper the higher state is consistently chosen as the surviving predecessor. Finally, the state with the least accumulated path metric at the current time instant is located. This state is called the global winner and is the state from which traceback operation will begin. This method of starting the trace back operation from the global winner instead of an arbitrary state was described by Linda Bracken bury [22] in her design of an asynchronous Viterbi decoder. This greatly improves probability of finding the correct traceback path quicker and hence reduces the amount of history information that needs to be maintained. It also reduces the number of updates required to the surviving path. Both these measures result in improved energy savings. The values for the surviving predecessors (also called local winners) and the global winner are passed to Block 3.

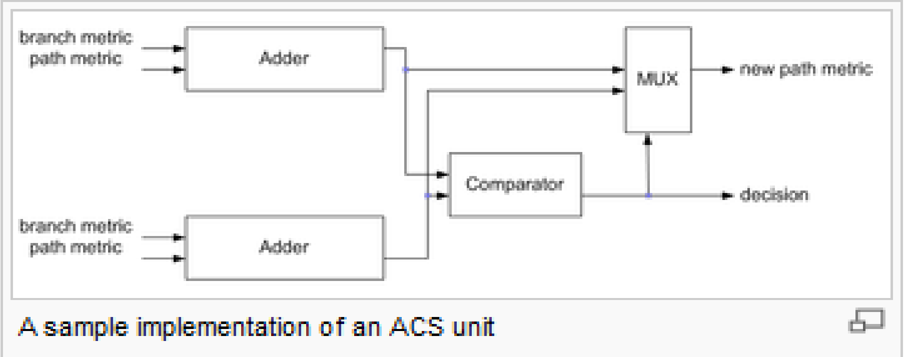


Fig -2.4 A sample implementation of an ACS unit

2.4 Traceback Unit

The global winner for the current state is received from Block 2. Its predecessor is selected in the manner . In this way, working backwards through the trellis, the path with the minimum accumulated path metric is selected. This path is known as the traceback path. A diagrammatic description will help visualize this process.the trellis diagram for a ½ K=3 (7, 5) coder with sample input taken as the received data.

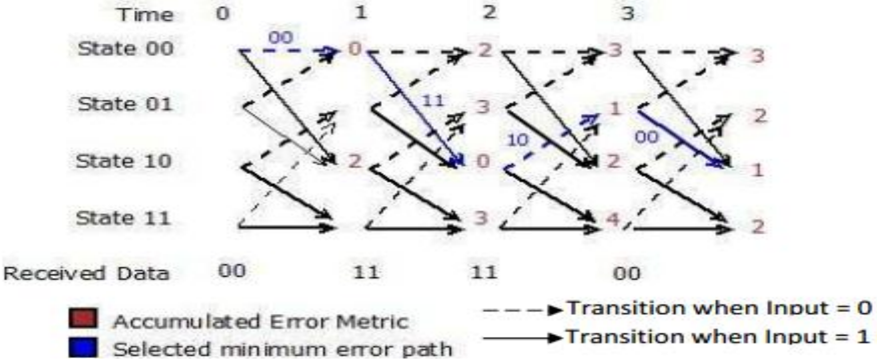


Fig -2.5 Selected minimum error path for a1/2 k=3(7,5) decoder

The state having minimum accumulated error at the last time instant is State 10 and traceback is started here. Moving backwards through the trellis, the minimum error path out of the two possible predecessors from that state is selected. This path is marked in blue. The actual received data is described at the bottom while the expected data written in blue along the selected path. It is observed that at time slot three there was an error in received data (11). This was corrected to (10) by the decoder.

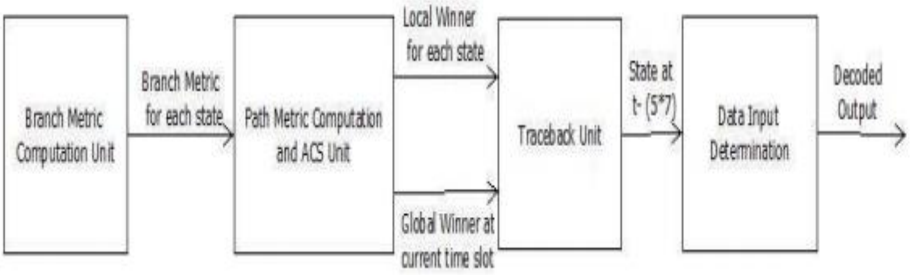

Branch Metric Computation Unit
Branch Metric for each state
Path Metric Computation and ACS Unit
Local Winner for each state
Global Winner at current time slot
Traceback Unit
State at t- (5*7)
Data Input Determination
Decoded Output

Fig-2.6 Schematic representation of the viterbi decoder

Local winner information must be stored for five times the constraint length. For a K =7 decoder, this results in storing history for 7 x 5 = 35 time slots. The state of the decoder at the time instant 35 time slots prior can then be accurately determined. This state value is passed to Block 4. At the next time slot, all the trellis values are shifted left to the previous time slot. The path metric for the last received data and compute the minimum error path is then calculated. If the global winner at this stage is not a child of the previous global winner, the traceback path has to be updated accordingly until the traceback state is a child of the previous state.

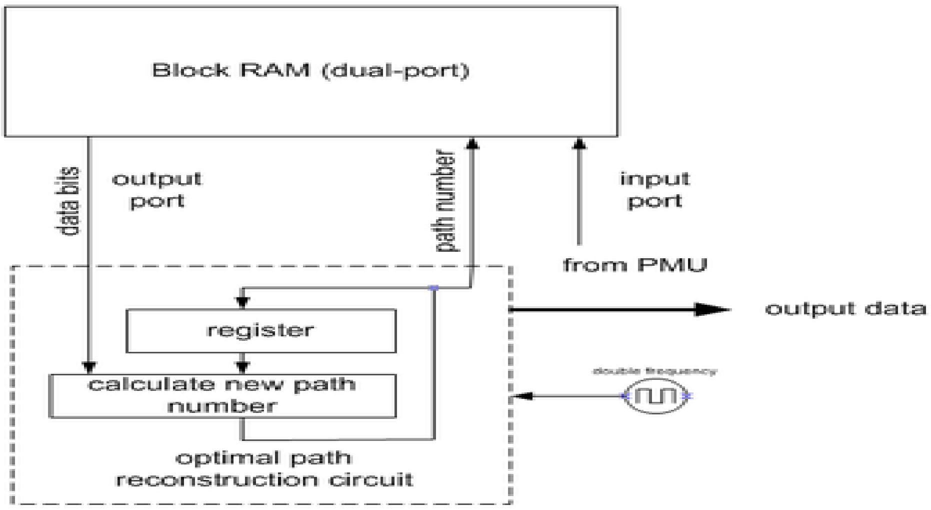


Fig-2.7 Block diagram of the Trace Back unit

Multiple traceback paths are possible and it may be thought that traceback up to the first bit is necessary to correctly determine the surviving path. However, it was found that all possible paths converge within a certain distance or depth of trace back. This information is useful as it allows the setting of a certain trace back depth beyond which it is neither necessary nor advantageous to store path metric and other information. This greatly reduces memory storage requirements and hence energy consumption of the decoder. Empirical observations showed that a depth of five times the constraint length was sufficient to ensure merging of paths . Therefore, local winner information is stored for 35 slots (five times seven) in the decoder used for this paper. Block 4. Data Input Determination Now going forwards through the traceback path, the state transitions at successive time intervals are studies and the data bit that would have caused this transition is determined. This represents the decoded output.

Determining Successors to a particular State Each state is represented by 6 shift registers (in the case of a K=7 encoder or decoder). The next state can therefore be obtained by a right shift of the values of the shift registers. The first shift register is given a value of 0. The resulting state represents the next state of the coder if the input bit was 0. By adding 32 (1x25) to this value, the next state of the coder if the input bit was 1 Determining Predecessors to a particular State In a similar way, the first predecessor can be calculated this time by a left shift of the values of the shift registers. By adding one (1x20) to this value, the value of the second predecessor to the state is derived.

2.5 State Metric Storage: The block stores the partial path metric of each state at the current stage.

2.6 Output Generator:

This block generates the decoded output sequence. In the traceback approach, the block incorporates combinational logic, which traces back along the survivor path and latches the path (equivalently the decoded output sequence) to a register.

2.7 Encoding Mechanism

Data is coded by using a convolutional encoder. It consists of a series of shift registers and an associated combinatorial logic. The combinatorial logic is usually a series of exclusive-or gates. The conventional encoder ½ K=7, (171,133) is used for the purpose of this paper. The octal numbers 171 and 133 when represented in binary form correspond to the connection of the shift registers to the upper and lower exclusive-or gates respectively. Figure 3.1 represents this convolutional encoder that will be used for the paper.

2.8 Decoding Mechanism

There are two main mechanisms by which Viterbi decoding may be carried out namely, the Register Exchange mechanism and the Traceback mechanism. Register exchange mechanisms, as explained by Ranpara and Sam Ha store the partially decoded output sequence along the path. The advantage of this approach is that it eliminates the need for traceback and hence reduces latency. However at each stage, the contents of each register needs to be copied to the next stage.

This makes the hardware complex and more energy consuming than the traceback mechanism. Traceback mechanisms use a single bit to indicate whether the survivor branch came from the upper or lower path. This information is used to traceback the surviving path from the final state to the initial state. This path can then be used to obtain the decoded sequence. Traceback mechanisms prove to be less energy consuming and will hence be the approach followed in this paper.

Decoding may be done using either hard decision inputs or soft decision inputs. Inputs that arrive at the receiver may not be exactly zero or one. Having been affected by noise, they will have values in between and even higher or lower than zero and one. The values may also be complex in nature.

In the hard decision Viterbi decoder, each input that arrives at the receiver is converted into a binary value (either 0 or 1). In the soft decision Viterbi decoder, several levels are created and the arriving input is categorized into a level that is closest to its value. If the possible values are split into 8 decision levels, these levels may be represented by 3 bits and this is known as a 3 bit Soft decision.

This paper uses a hard decision Viterbi decoder for the purpose of developing and verifying the new energy saving algorithm. Once the algorithm is verified, a soft decision Viterbi decoder may be used in place of the hard decision decoder. Figure 3.2 shows the various stages required to decode data using the Viterbi Algorithm. The decoding mechanism comprises of three major stages namely the Branch Metric Computation Unit, the Path Metric Computation and Add-Compare-Select (ACS) Unit and the Traceback Unit.

3.Main Objectives: The main objectives of this paper are as follows

1. An understanding of the background literature relevant to error detection and error control mechanisms as currently used in packetized digital communication networks.
2. A detailed understanding of the concept of convolutional coding, and decoding using the Viterbi algorithm.
3. An implementation of the Viterbi algorithm in verilog so it is working correctly by comparing its performance with that of the Viterbi decoder function provided by verilog (A designed Viterbi decoder is needed because verilog does not provide access to the code.
4. A resolution of questions that still need to be answered about the T- algorithm including the correct initialization of component decoders and the stability of the feedback mechanism
5. An implementation in verilog of the T- algorithm as a modification of the Viterbi algorithm.
6. An evaluation of the T- algorithm in terms of its accuracy and capacity for achieving energy saving Timing Analysis will be performed on the basis of bit-error performance, packet loss rates and execution time (considered to provide a first order approximation to energy consumption).

III. RESULTS

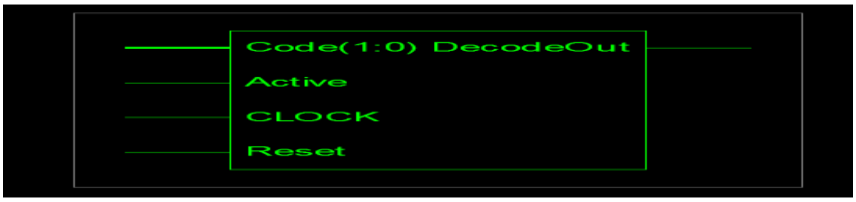


Fig-4.1 Top Module of VITERBI DECODER

In viterbi decoder it has 4 modules BMU,PMU,ACSU,SMU where the input is taken by the BMU and output arrives at SMU

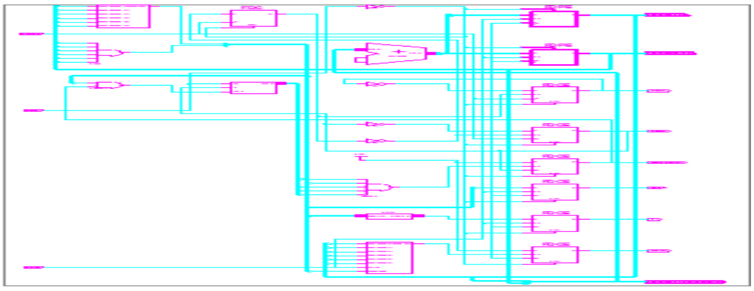


Fig-4.2Control Unit SCHEMATIC DIAGRAM

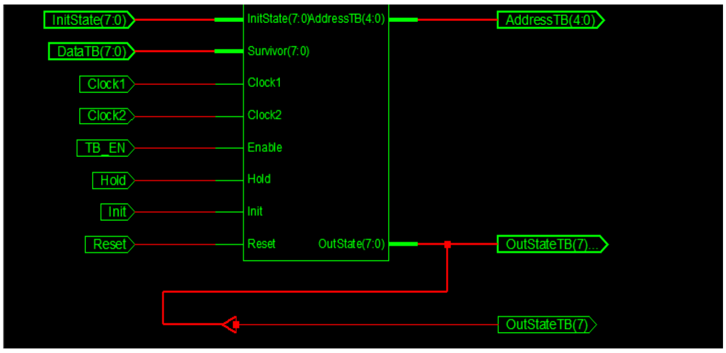
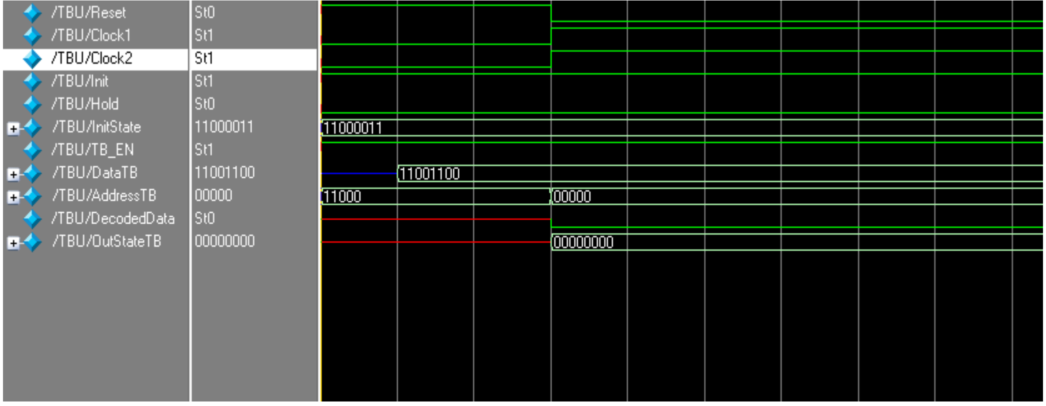


Fig-4.3 TRACEBACK UNIT BLOCK DIAGRAM

In trace back unit the input will be taken from the add compare select unit and again it will trace back to the path metric whether it is the shortest path are not it calculate and it will send to the survivor metric unit



```
/TBU/Reset        St0
/TBU/Clock1       St1
/TBU/Clock2       St1
/TBU/Init         St1
/TBU/Hold         St0
/TBU/InitState    11000011
/TBU/TB_EN        St1
/TBU/DataTB       11001100
/TBU/AddressTB    00000
/TBU/DecodedData  St0
/TBU/OutStateTB   00000000
```

Fig-4.4 TRACEBACK UNIT WAVEFORM

Reset-input;Clock1-input;Clock2-input;Lnit-input;Hold-input;Data_TB-output

IV. CONCLUSION

We have proposed a high speed low power VD design for TCM systems.theprecomputation architecture that incorporates T-algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. we have also analysed precomputation algorithm .where the optimal precomputation steps are calculated and dicussed.this algorithm is suitable for TCMsystems which always employ high rate convolutional code.finally we presented a design case.Both the ACSU and SMU are modified to correctly to decode the signal FPGA and power estimation results show that ,compared with the full trellis VD without a low power scheme,theprecomputation VD could reduce the power consumption by 70% with only 11% reduction of the maximum decoding speed.

By using FPGA device and hybrid microprocessor the decoding benefits can be achieved in future. In future to improve the decoder performance the Viterbi algorithm is carried out in reconfigurable hardware. power saving architecture can be designed for the above decoder which is executable in the mobile devices. Viterbi decoder can also be implemented using verilog. Therefore in the future Viterbi algorithm may be used for various scenarios. So in the future the complexity can be greatly reduced. By using FPGA device and hybrid microprocessor the decoding benefits can be achieved in future. In future to improve the decoder performance the Viterbi algorithm is carried out in reconfigurable hardware. power saving architecture can be designed for the above decoder which is executable in the mobile devices. Viterbi decoder can also be implemented using verilog. Therefore in the future Viterbi algorithm may be used for various scenarios. So in the future the complexity can be greatly reduced.

REFERENCES

[1]. The "20 Mbps convolutional encoder Viterbi decoder STEL-2020,"*Stanford Telecommunications, Santa Clara, CA*, Oct. 1989.

[2]. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of computer Algorithms* Reading, MA Addison-Wesley, 1974.

[3]. ATSC Standard a/53(1995) "ATSC digital television standard", 1995.

[4]. P.J. Black and T.H.Meng, "A hardware efficient parallel Viterbi algorithm," in *ICASSP* , vol.2, pp. 893-896, 1990.

[5]. 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.

[6]. P.J. Black and T.H. Meng," A unified approach to the Viterbi algorithm state metric update for shift register processes," *in Proc. ICASSP*, vol.5, pp. 629-632, Mar. 1992.

[7]. S. Benedetto, G. Montorsi, "Unveiling Turbo codes: Some results on parallel concatenated coding schemes," IEEE Transactions on Information Theory, Vol. 42 No. 2, p. 409-428, March 1996.

[8]. A. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 473-484, Apr. 1992.

[9]. R. Cypher and C.B. Shung, "Generalized traceback techniques for survivor memory management in the Viterbi algorithm," in *Proc.* GLOBECOM, pp. 1318-1322, Dec. 1990.

[10]. J.E. Dunn,"A 50Mb/s multiplexed coding system for shuttle communications," *IEEE trans. Commun.*, vol. COM-26, pp. 1636-1638, Nov. 1978.

[11]. G Fettweis and H, Meyr, "High-speed parallel Viterbi decoding algorithm and VLSI-architecture," *IEEE communication Magazine*, vol. 29, pp. 46-55, May 1991.

[12]. G. Fettweis, H. Meyr, "A 100 Mbits/s Viterbi decoder chip: Novel architecture and its realization," *IEEE ICC'90,* 307.4, 463-467, April 1990.

[13]. G. Fettweis, H. Meyr, "A modular variable speed Viterbi decoding implementation for high data rates," *North-Holland: Signal Processing IV, Proc. EUSIPCO'88*, 339-342, 1988.

[14]. G. Fettweis, H. Meyr,"A systeolic array Viterbi processor for high data rates," *Int. Conf. On Systolic Arrays, Ireland*, 1989, Prentice Hall: '*Systolic Array Processors*', 195-204,1989.