International Journal of Academic Research and Development
ISSN: 2455-4197
Impact Factor: RJIF 5.22
www.academicsjournal.com
Volume 3; Special Issue 2; March 2018; Page No. 56-59

Implementation of TCM encoder and decoder for forward error correction

Kshitij Saratkar, Sachin Meshram, Vandana Bade
Professor, Electronics Engineering, SPPU, GHRCEM, Ahmednagar, Maharashtra, India

Abstract
This paper presents the technique to design the area efficient trellis code modulation decoder and encoder. The best encoding technique is convolutional encoding as convolutional encoding improves the error performance of communicational channel. But this will increase the bandwidth requirement of system. The trellis code modulation (TCM) can achieve the better error performance without Bandwidth expansion or extra power.
This technique is aims to reduce the hardware requirement of the TCM decoder by simply using the Look-Up Tables (LUTs) in the design. In this technique we will directly use the Look-Up tables (LUTs). There are two LUTs use in this technique, and they are hamming distance look-up table (HDLUT), and Output look-up table (OLUT). Instead of designing a circuit to compute the hamming distance in decoding algorithm i.e. viterbi decoding, we pre-calculating the hamming distance and store in the look-up table. Another look-up table use is the output look-up table (OLUT) which gives the output of the viterbi decoder.

Keywords: look-up table (LUT), trellis code modulation (TCM), viterbi decoding

1. Introduction
Trellis code modulation (TCM) is combine technique use for both modulation and error correcting in digital communication. It's aim is to achieve the better error performance without Bandwidth expansion or extra power. In the case of TCM, encoder use is convolutional encoder as it provides the forward error correction, and decoder use to decode and recover data at receiver end is Viterbi decoder (i.e., the maximum likelihood decoder). This algorithm involves large amount of computation and storage as it involves calculating the branch costs of a survivor path, but here in this technique to reduce the hardware requirement of the TCM decoder by simply using the Look-Up Tables (LUTs) in the design. In this technique we will directly use the Look-Up tables (LUTs). There are two LUTs use in this technique, and they are hamming distance look-up table (HDLUT), and Output look-up table (OLUT). Instead of designing a circuit to compute the hamming distance in decoding algorithm i.e. viterbi decoding, we pre-calculating the hamming distance and store in the look-up table. Another look-up table use is the output look-up table (OLUT) which gives the output of the viterbi decoder. We will design trellis code modulation encoder and decoder by using HDL such as VHDL or Verilog.

2. TCM Encoder Architecture
Figure shows the proposed architecture of the TCM encoder. TCM encoder is mainly consisting of convolutional encoder, which is use for the forward error correction in the communicational channel. Convolutional encoder is also known as trellis encoder. In this method coding gain is improve by simply increasing the density of constellation while keeping the minimum distance between each constellation point same. QAM is the method use to modulate the digital data in to the analog signal.

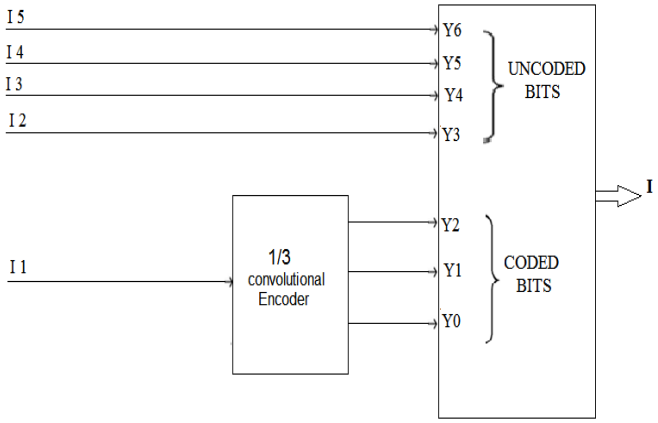

Fig 1: TCM Encoder [1]

Figure shown is TCM encoder which consists of five inputs and seven output. Main part of a TCM encoder is 1/3 convolutional encoder, which have one bit input and three bits output. This 3 bit output is depending on the current state and input applied to the encoder. Let say input (I1) is applied to the encoder which gives the outputs (Y0, Y1, Y2) which depend on the current state and input applied to the encoder. Convolutional encoder is consist of shift register when we apply one bit data to the convolutional encoder at positive edge of the clock, at the same time data in the shift register is shifted by one position and it will give the three bit output[1].
In this way five bit data is encoded in to seven bits in TCM encoder. which then modulated by QAM modulation in which each combination of phase and amplitude is one of the 2^5 for 5-bit patterns. After a convolutional encoding, each symbol has 7 bits, which requires a 2^7 point constellation. So

distance between constellation points is reduces and so error occurs in the transition channel are reduce.

3. TCM Decoder Architecture

Parallel processing is employing at every state, to trace the history delays in each state. Working of the TCM decoder is mainly based on the viterbi algorithm. It consist of three parts state-transition and add unit (STAU), the compare-select unit and trace back unit.

Output of the TCM Encoder which is seven bit is applied to the TCM decoder. At the first stage seven bits are splits in to the two part i.e. lower three bits and upper four bits. Lower 3 bits are applied to the viterbi decoder. There are eight states in the decoder (i.e. from "000" to "111"). The first part of the decoder is STAU, where we apply the current state which splits in to two output which is consist of next state and path cost. As shown in the block diagram there are eight STAU for state from "000" to "111". Every STAU have two outputs (i.e. after application 0 and 1)

Output of STAU is given to the CS2U (i.e. compare select 2 unit) which select a path with minimum cost. Output of CS2U is apply to the CS8U which is use to select the survivor path. And next states are simply store in the shift registers which is use for back tracing. Portion of Output of the CS2U that consist of path cost is apply to the STAU as a feedback, which then add to next path cost to give branch cost in next state. Output of CS2U is apply to the CS8U, which finds the survivor path (i.e. path with minimum path cost).

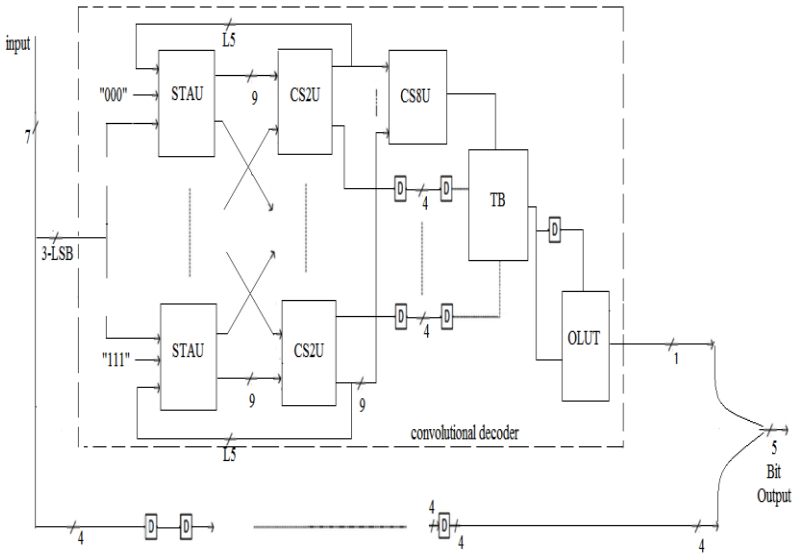

Fig 2: TCM Decoder [1]

Later Survivor Memory Unit is used which stores decoded bits of survivor paths. Various methods are used to retrieve data of survivor path with lower path metric and they are Trace back method and register exchange method. But in this technique in order to improve the area and performance we implementing the register exchange method [2]. And after that from OLUT we get the output if the viterbi decoder which is one bit output. In this way this one bit and remaining four bits of input are combines to give five bit output of the TCM Decoder.

3.1 Construction of look up table (LUT)

In this technique to reduce the hardware requirement of the TCM decoder by simply using the Look-Up Tables (LUTs) in the design. In this technique we will directly use the Look-Up tables (LUTs). There are two LUTs use in this technique, and they are hamming distance look-up table (HDLUT), and Output look-up table (OLUT). Instead of designing a circuit to compute the hamming distance in decoding algorithm i.e. viterbi decoding, we pre-calculating the hamming distance and store in the look-up table. Another look-up table use is the output look-up table (OLUT) which gives the output of the viterbi decoder [1]

HDLUT which is used in design of STAU is constructed from trellis diagram of convolutional decoder. Here we pre-calculate the hamming distance and store in the look-up table. Hamming distance is nothing but the difference of received bits and actual bit present at that position.

Table 1: Hamming Distance Table [1]

True path state	Corresponding hamming distance							
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
000 (P0)	0	1	1	2	1	2	2	3
001 (P1)	1	0	2	1	2	1	3	2
010 (P2)	1	2	0	1	2	3	1	2
011 (P3)	2	1	1	0	3	2	2	1
100 (P4)	1	2	2	3	0	1	1	2
101 (P5)	2	1	3	2	1	0	2	1
110 (P6)	2	3	1	2	1	2	0	1
111 (P7)	3	2	2	1	2	1	1	0

Each path state is 3-bit binary data of "000" to "111" for the 1/3 convolutional encoder. In the TCM convolutional decoder, the input symbol should be one of eight possible three coded bits from "000" to "111". Hamming distance is easy to calculate. This distance is computed by simply counting how many bits differ between the received symbol pairs and the possible symbol pairs. The distances are pre-calculated and

compiled into Table. The OLUT is place in to the pseudo ROM. OLUT directly gives the output of the convolutional decoder.

4. Convolutional Encoder Architecture

TCM encoder is nothing but the convolutional encoder, here convolutional encoder use is 1/3 convolutional encoder. which have one bit input and three bits output. This 3 bit output is depending on the current state and input applied to the encoder. Let say input (I1) is applied to the encoder which gives the outputs (Y0, Y1, Y2) which depend on the current state and input applied to the encoder. Convolutional encoder is consist of shift register when we apply one bit data to the convolutional encoder at positive edge of the clock, at the same time data in the shift register is shifted by one position and it will give the three bit output.

This is the structure of the proposed convolutional encoder having constraint length of 5. And output of convolutional encoder is given by

Y0 = X0 (XOR) X1
Y0 = X1 (XOR) X2
Y0 = X0 (XOR) X2

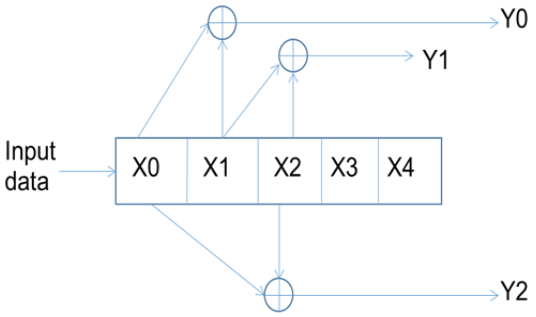

Fig 3: Convolutional Encoder.

5. Results

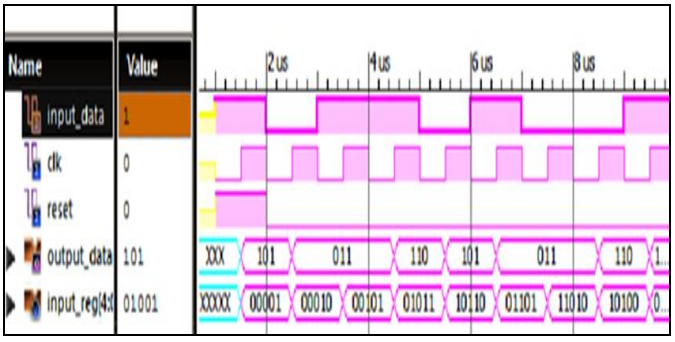

Fig 4: Simulation Result of convolutional Encoder

This is the simulation result of 1/3 convolutional encoder, in which input data is random bits. When reset is set to logic one, input register which is shift register is reset to "00000" and it take the value of input data at every positive edge of clock, and output is depend on the content of the input register.

So our proposed architecture will reduce the area and power as compare to the conventional methods. So it is area efficient method of implementation of TCM decoder.

Viterbi Decoder

Device Utilization Summary of the viterbi decoder is given in Table III. Were VHDL code is use for designing the viterbi decoder, and Xilinx 13.2 design software is use for simulation where Targeted device used is XC6SLX16 which belongs to family Spartan-6.

Table 2: Device utilization summary

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	35	18224	0%
Number of Slice LUTs	1306	9112	14%
Number of fully used LUT-FF pairs	30	1311	2%
Number of bonded IOBs	173	232	74%
Number of BUFG	1	16	6%

RTL schematic of viterbi decoder using hybrid register exchange method is given in fig 5.

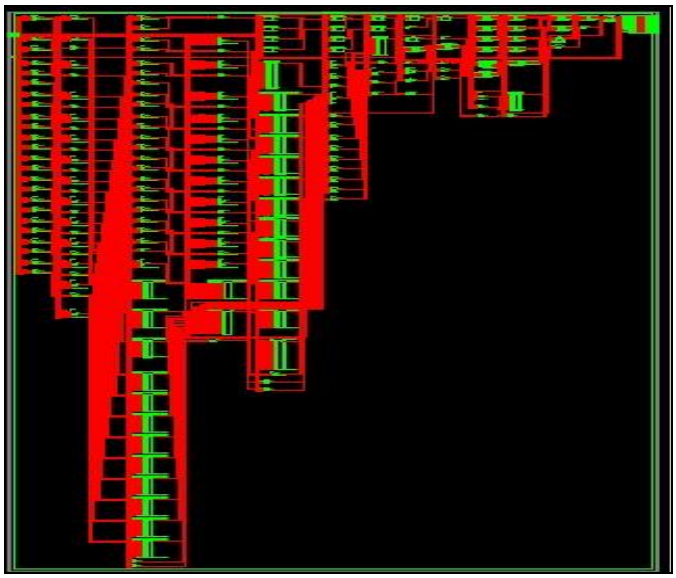

Fig 5: RTL schematic of Viterbi Decoder

6. Conclusion

This paper proposed an improved structure of TCM decoder, in which we are using Hamming distance look-up table (HDLUT), and output look-up table (OLUT). So as to reduce the area and power. And register exchange method is use for the back tracing of survivor path so as to reduce the memory requirement. So proposed TCM decoder is efficient in area.

7. References

1. Anh Dinh, Xiao Hu. A Hardware-Efficient Technique to Implement a Trellis Code Modulation Decoder, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2005, 13(6).
2. Dalia A El-Dib. Memoryless Viterbi Decoder, IEEE Transactions on Circuits and Systems, 2005, 52(12).
3. Sunil P Joshi, Roy Paily. Low Power Viterbi Decoder by Modified ACSU architecture and Clock Gating Method" IEEE Int. Conf, 2011, 499-503.
4. Viterbi AJ. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Inform. Theory. 1967; 13(2):260-269.

5. Joeressen OJ, Meyr H. A 40 Mb/s soft-output Viterbi decoder, IEEE J. Solid-State Circuits. 1995; 30(7):812-818.
6. Wei LF. Trellis-coded modulation with multidimensional constellations, IEEE Trans. Inform. Theory. 1987; 33(4):483-501.
7. Gierenz VS, Weiss O, Noll TG. A 550 Mb/s radix-4 bit-level pipelined 16-state 0.25- m CMOS Viterbi decoder, in Proc. IEEE Int. Conf. Application-Specific Systems, Architectures, and Processors, Boston, MA. 2000; 10(12):195-201.