

Space-time trellis codes: Field programmable gate array approach

Mallikarjuna Gowda C. P., Raju Hajare

Department of Electronics and Telecommunication Engineering, BMS Institute of Technology and Management, India

Article Info

Article history:

Received Feb 18, 2020
Revised Mar 19, 2020
Accepted May 17, 2020

Keywords:

Digital signal processors
FPGA
Space-time trellis codes
Squared euclidean distance
Viterbi decoding algorithm

ABSTRACT

In this paper, an effective method of implementation of Space-Time Trellis Codes (STTC) for 4-state on Field Programmable Gate Array is presented. To reach the very high data rates provided in STTC, a lot of expensive high-speed Digital Signal Processors (DSPs) should be employed for the real time applications, while it might not be affordable. This fact has motivated in designing dedicated hardware implementations using Field Programmable Gate Array (FPGA) with low cost and power consumption. The hardware device XC3S400, family Xilinx Spartan-3, and package PQ208 are used in this project, in which the STTC encoder and decoder utilizes maximum 10% and 22% as that of available device capacity respectively. The design has been simulated and synthesized successfully in Xilinx integrated software environment.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mallikarjuna Gowda C. P.,
Department of Electronics and Telecommunication Engineering,
B M S Institute of Technology and Management,
Doddaballapur Main Road, Avalahalli, Yelahanka, Bengaluru, Karnataka 560064, India.
Email: cpmallikarjunagowda@bmsit.in

1. INTRODUCTION

Space-Time Trellis Coding (STTC) is a channel coding technique that can be used to improve the performance of wireless communication systems over fading channels. STTC combines both space and time diversity to avoid multipath fading. Several researchers have undertaken the construction of space-time trellis codes [1]. The rank and determinant criterion (RDC) and Euclidean distance criterion (EDC) have been developed as design criteria. The space-time (ST) coding can make good use of the transmit diversity to alleviate the effect of fading and achieve high data transmission rate without adding extra bandwidth and power consumption. The first ST code with a normalized rate of 1 symbol per channel use (pcu) was proposed by Alamouti over two transmit antennas and two time periods [2]. In [3], Tarokh gave the design criteria, which is a tradeoff between constellation size, data rate, complexity, and diversity advantage for ST code; he presented space-time trellis code (STTC) is able to combat the effect of fading, which can simultaneously offer a substantial coding gain, spectral efficiency, and diversity improvement on flat fading channels [4-10].

2. HARDWARE IMPLEMENTATION

The hardware implementation of STTC encoder for any number of states is less complex and for more number of states in the decoder the implementation complexity of hardware increases. The Viterbi algorithm to perform maximum-likelihood decoding is attractive for efficient VLSI hardware implementation.

2.1. STTC encoder

Space-time trellis codes (STTCs) combine modulation and trellis coding to transmit information over multiple transmit antennas [1]. To achieve full diversity, a STTC should satisfy the rank criterion where as to achieve full data rate, should satisfy the determinant criterion. Let us assume a constellation that maps c bits of data to a symbol, the delay diversity scheme uses every c input bit to pick one symbol that is transmitted from the second antenna. Then, the first antenna transmits the same symbols with a delay of one symbol. Since the symbols go through two independent channel path gains, a diversity of two is achieved for one or more receive antennas.

The block diagram of STTC encoder is shown in Figure 1. In our implementation, we considered a 4-state space-time trellis coded QPSK scheme with 2-transmit antennas. The encoder consists of a bit splitter, two feedforward shift registers, four multipliers, and modulo 4-adder. A bit splitter splits the input sequence, c , into two parallel bit streams which are delayed by one bit delay circuit. The bits, c_t^1 & c_t^2 , and the delayed bits, c_{t-1}^1 & c_{t-2}^2 , are multiplied with an encoder coefficient set and hence, the results of all the multipliers are added by modulo-4 adder. The adder outputs, x_t^1 & x_t^2 , are points from a QPSK constellation; they are transmitted simultaneously through the 1st and 2nd antenna, respectively.

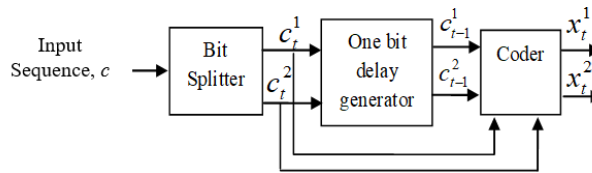


Figure 1. Block diagram of STTC encoder

2.1.1. Bit splitter

The Circuit diagram of Bit Splitter as illustrated in Figure 2. The input sequence, c , is applied to the flip-flop-1, which output is given as input to flip-flop-2 with a common clock. A flip-flop-3 is a mod-2 counter which output is used as a clock for the flip-flop-4 & 5. Initially, all the flip-flops are reset to zero which are triggered at negative edge clock. During first clock cycle the first LSB of data is stored in the flip-flop-1 while flip-flop-1 output is stored in the flip-flop-2 and the flip-flop-3 generates the logic '1' output. For next clock cycle the first LSB of data is stored in the flip-flop-2 while second LSB of data will be in flip-flop-1 and the flip-flop-3 generates logic '0'. So, in this instant the flip-flop-4 & 5 will simultaneously be triggered and the flip-flop-4 gives first LSB of data whereas the flip-flop-5 gives second LSB of data.

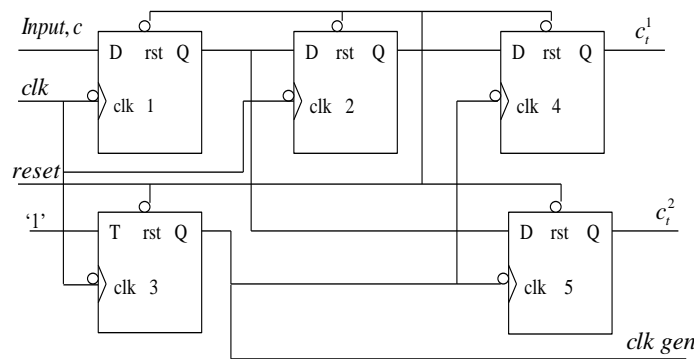


Figure 2. Circuit diagram of bit splitter

2.1.2. Delay generator

The Circuit diagram for Delay generator as illustrated in Figure 3. The state selection is taken for 64-state but we have done for only 32-state, and explanation with an example is given for 4-state only.

However, six delay elements required for 64-state in which three delay elements are used in upper line & another three delayed elements are in lower lines. Figure 3, shows the circuit diagram of delay generator, which consists of three D flip-flops are connected in series in upper line where as another three D flip-flops are connected in series in lower line. The upper line flip-flops are required to generate one-bit delay for the data, whereas the lower line flip-flops are used to generate one-bit delay for the data.

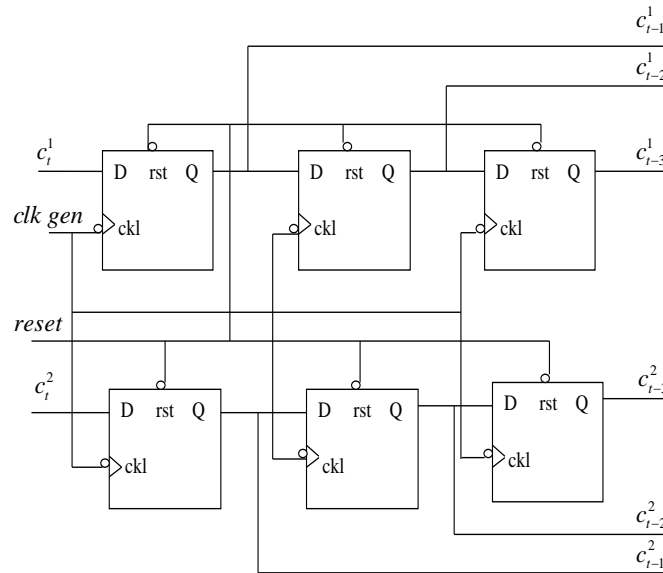


Figure 3. Circuit diagram for delay generator

2.1.3. Coder

A coder consists of multiplier, state selector, and modulo-4 adder. The internal architecture (RTL Schematic) of coder is shown in the appendix. If an encoder which is operated with either 4-state or 8-state or 16-state or 32-state, the state selector should select either 010 or 011 or 100 or 101 respectively. The encoder coefficient set are multiplied with input bits and delayed bits by multiplier and then modulo-4 adder is used to add all the multiplier outputs, since the encoded symbols 0, 1, 2, & 3 are used to modulate for QPSK modulator. The adder outputs x_t^1 & x_t^2 are points from a QPSK constellation which are transmitted simultaneously through the first and second antenna, respectively.

2.2. Algorithm for STTC encoder

Step 1: Clear all the memory elements (shift registers) to zero, say $d1=0$; $d2=0$.

Step 2: Input sequence, c , is splitted into two streams of bits, say c_t^1 & c_t^2 .

Step 3: Bit streams, c_t^1 & c_t^2 , and bits from all shift registers are multiplied by 1st transmit antenna encoder coefficient set. The STTC symbol for 1st transmit antenna, say x_t^1 , is obtained by the multiplier outputs from all shift registers are modulo-4 added.

Step 4: Bit streams, c_t^1 & c_t^2 , and bits from all shift registers are multiplied by 2nd transmit antenna encoder coefficient set. The STTC symbol for 2nd transmit antenna, say x_t^2 , is obtained by the multiplier outputs from all shift registers are modulo-4 added.

Step 5: Bit streams, c_t^1 & c_t^2 , are delayed by one bit and stored in the shift registers.

Step 6: For 2nd time, 3rd time, ... repeats the steps from step 2 to 5.

2.3. Flow chart for STTC encoder

The flowchart for STTC encoder is shown in Figure 4, as a first step all the shift registers are cleared, i.e., reset to zero, then the input sequence is splitted into to two streams of bits, i.e., c_t^1 & c_t^2 .

The STTC symbols for first and second transmit antennas x_t^1 & x_t^2 , are computed by the multiplier outputs from all the shift registers are modulo 4 added. Bit streams c_t^1 & c_t^2 are delayed by one bit and stored in the shift registers. This process is repeated till the input exists.

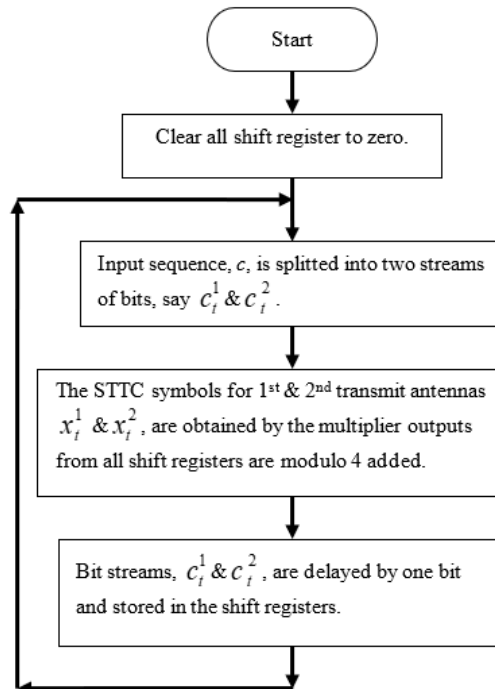


Figure 4. Flowchart for STTC encoder

2.4. STTC decoder

The implementation of STTC Decoder on FPGA is shown in Figure 5 which consists of code converter, Squared Euclidean Distance (SED) generator, feasible state selector, feasible branch selector, PISO converter, and Detector. The received symbols from receive antennas are selected from the diversity selector and demodulated from the QPSK demodulator. The demodulated outputs are in the form of binary bits, which are converted to decimal symbols from the code converter. These symbols are given to the squared Euclidean distance generator, which generates the squared Euclidean distances for all states. The squared Euclidean distances are calculated for each state which contains 4 branches and hence the squared Euclidean distance is to be calculated for all sixteen combinations for 4-state. The SEDs for all states are fed into the feasible state selector, which selects the required states. And it counts the number of states are selected and gives the corresponding Most Significant Digit (MSD) for each state. The required state is that the SEDs of a state of its branches are less than or equal to 2.

In 4-states STTC, maximum three required states are obtained by the feasible state selector. These states are given to the three feasible branch selectors. The feasible branch selector selects the required branches from each state. Required branch is that the SED of the branch of a state is less than or equal to 2. And feasible branch selector gives the corresponding Least Significant Digit (LSD) for each branch and gives the SEDs for required branches in a state are given to the detector. The detector consists of reference symbol generator, comparator, subtractor, and decoder. The reference symbols are initially set to zero. The squared Euclidean distance for the reference symbols are calculated with its reference state (MSD of reference symbols) from the SED generator. The feasible branches are selected from the values of SEDs for the reference symbols. The feasible branch selector gives the SEDs and LSDs for feasible branches.

The detector compares the all SEDs which are taken from the three feasible branch selectors for states and the reference symbols. And it selects any one survive (likely) path, feasible reference symbols, and required output data. The output data is in the form of decimal digit, which is converted into binary from the code converter. These binary bits are in the form of parallel bits which are converted into serial data. The serial data is called information.

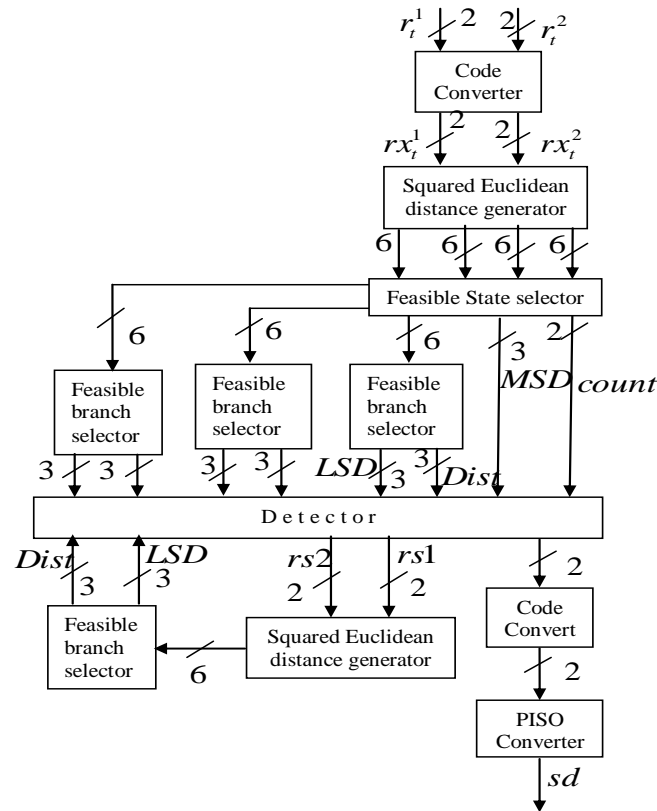


Figure 5. Components for STTC decoder

2.5. Algorithm for STTC decoder

- Step 1: Clear all the reference symbols, states, branch metrics for all the 16 combinations and reference symbol, memory elements which are used to store and processing the symbols to zero.
- Step2: Received symbols, r_i^1 and r_i^2 are converted into decimal symbols, rx_i^1 and rx_i^2 , respectively.
- Step 3: Find SEDs for all the 16 combinations and the results are stored into memory.
- Step 4: Select the feasible states from above step, say, S_0' , S_1' and S_2' .
- Step 5: Select feasible branch metrics for each feasible state and the results are stored into memory.
- Step 6: Find SEDs for the reference symbols, dl and dr and the results are stored into memory.
- Step 7: Select feasible branch metrics from above step. The results are stored into the memory.
- Step 8: Subtract all the LSD of feasible branches of reference symbol from the input symbol, rx_i^1 (MSD).
- Step 9: Compare them and select least result which was generated for the LSD of reference symbol. It represents the present state among the four states.
- Step10: Select the symbol that was least SED among four branches in the state. These symbols are reference symbols for the next state.
- Step 11: Select the MSD of reference symbol is the output symbol.
- Step 12: The output symbol is converted into binary, which is converted back to serial data which is the actual information.
- Step 13: Repeat the steps from 2 to 12 for next receive data.

2.6. Flow chart for STTC decoder

The flow chart for STTC decoder is shown in Figure 6, as a first step, clear all the reference symbols, states, branch metrics and memory elements to zero. To process the received symbols easily, convert them into decimal symbols. Calculate the SEDs for all the sixteen combinations and reference symbols, then select the feasible states and branches from the four states and also select the feasible branches for reference symbols. Subtract all the LSD of feasible branches of reference symbol from the MSD of input received symbols. Determine the present state, reference symbol and output. The output is to serial data. This process is repeated till the completion of all states.

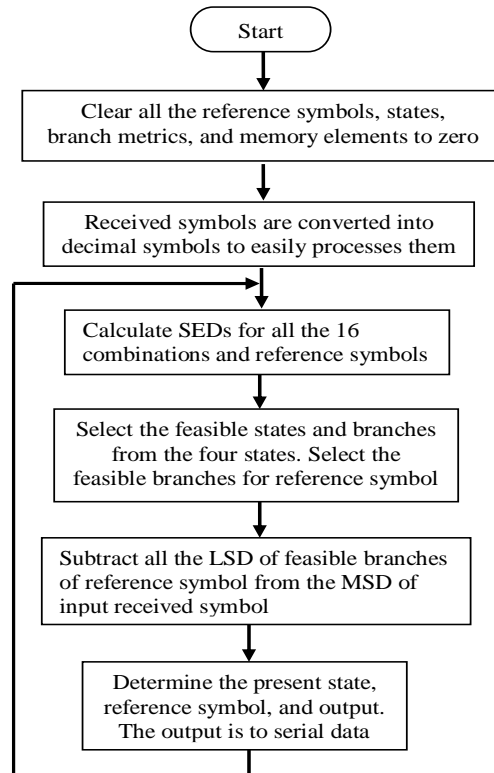


Figure 6. Flowchart for STTC decoder

3. METHODOLOGY

Let the transmitted signal sequences from the two transmit antennas are

$$x_t^1 = \{0, 2, 1, 3, 0, 2, 1, 3, \dots\}$$

$$x_t^2 = \{2, 1, 3, 0, 2, 1, 3, 0, \dots\}$$

and received sequences from the two receive antennas are

$$r_t^1 = \{0, 3, 0, 3, 0, 1, 2, 3, \dots\}$$

$$r_t^2 = \{1, 2, 2, 1, 3, 0, 3, 1, \dots\}$$

Let at time, $t = 0$, the reference symbols, d_l (MSD) and d_r (LSD) are initially 0 and 0, respectively and the received symbols r_t^1 (MSD) and r_t^2 (LSD) are 0 and 1, respectively. The SED for reference symbols (0, 0) and received symbols (0, 1) are shown in Table 1.

Table 1. Calculation of SEDs for $t = 0$.

0,0	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
0,0	0	0,1	0,0	1	1,0	2	2,0	5	3,0	10
0,1	1		0,1	0	1,1	1	2,1	4	3,1	9
0,2	4		0,2	1	1,2	2	2,2	5	3,2	10
0,3	9		0,3	4	1,3	5	2,3	8	3,3	13

The branches and states, which are having high SEDs (≥ 4), are eliminated. The absolute values is that the subtractor subtracts all the LSD (d_r) of reference symbol state from the MSD (r_t^1) of received symbol, ($r_t^1 - d_r$) is $abs(0 - 0) = 0$, & $abs(0 - 1) = 1$. The least difference value from reference symbol 0 is 0. Therefore, reference for the next state is 0 and the symbol which has least SED in the state-0 (S_0) is (0, 1). The next reference symbol is (0, 1) and output is **0**. For time, $t = 1$, the reference symbols, (0, 1) and the received symbols (3, 2) are shown in Table 2.

Table 2. Calculation of SEDs for $t = 1$

0,1	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
0,0	1	3,2	0,0	13	1,0	8	2,0	5	3,0	4
0,1	0		0,1	10	1,1	5	2,1	2	3,1	1
0,2	1		0,2	9	1,2	4	2,2	1	3,2	0
0,3	4		0,3	10	1,3	5	2,3	2	3,3	1

The branches and states, which are having high SEDs (≥ 4), are eliminated. The absolute values of $r_t^1 - d_r$ are 3, 2 and 1. The least difference value from reference symbol 2 is 1. Therefore, reference for the next state is 2 and the symbol which has least SED in the state-2 (S_2) is (2, 2). The next reference symbol is (2, 2) and output is **0**. For time $t = 2$, the reference symbols, (2, 2) and the received symbols, (0, 2) are shown in Table 3.

Table 3. Calculation of SEDs for $t = 2$

2,2	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
0,0	4	0,2	0,0	4	1,0	5	2,0	8	3,0	13
0,1	1		0,1	1	1,1	2	2,1	5	3,1	10
0,2	0		0,2	0	1,2	1	2,2	4	3,2	9
0,3	1		0,3	1	1,3	2	2,3	5	3,3	10

The branches and states, which are having high SEDs (≥ 4), are eliminated. The absolute values of $r_t^1 - d_r$ are 1, 2 and 3. The least difference value from reference symbol 1 is 1. Therefore, reference for the next state is 1 and the symbol which has least SED in the state-1 (S_1) is (1, 2). The next reference symbol is (1, 2) and the output is **2**. For time $t = 3$, the reference symbols, (1, 2) and the received symbols, (3, 1) are shown in Table 4.

Table 4. Calculation of SEDs for $t = 3$

1,2	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
1,0	4	3,1	0,0	10	1,0	5	2,0	2	3,0	1
1,1	1		0,1	9	1,1	4	2,1	1	3,1	0
1,2	0		0,2	10	1,2	5	2,2	2	3,2	1
1,3	1		0,3	13	1,3	8	2,3	5	3,3	4

The branches and states, which are having high SEDs (≥ 4), are eliminated. The absolute values of $r_t^1 - d_r$ are 2, 1 and 0. The least difference value from reference symbol 3 is 0. Therefore, reference for the next state is 3 and the symbol which has least SED in the state-3 (S_3) is (3, 1). The next reference symbol is (3, 1) and the output is **1**. For time $t = 4$, the reference symbols, (3, 1) and the received symbols, (0, 3) are shown in Table 5.

Table 5. Calculation of SEDs for $t = 4$

3,1	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
3,0	4	0,3	0,0	9	1,0	10	2,0	13	3,0	18
3,1	1		0,1	4	1,1	5	2,1	8	3,1	13
3,2	0		0,2	1	1,2	2	2,2	5	3,2	10
3,3	1		0,3	0	1,3	1	2,3	4	3,3	9

The branches and states, which are having high SEDs (≥ 4), are eliminated. The absolute values of $r_t^1 - d_r$ are 0 and 1. The least difference value from reference symbol 0 is 0. Therefore, reference for the next state is 0 and the symbol which has least SED in the state-0 (S_0) is (0, 3). The next reference symbol is (0, 3) and the output is **3**. Similarly, for time $t = 5, 6$ and 7 the reference symbols and the received symbols are shown in Table 6, Table 7 and Table 8.

Table 6. Calculation of SEDs for t = 5

0,3	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
0,0	9	1,0	0,0	1	1,0	0	2,0	1	3,0	4
0,1	4		0,1	2	1,1	1	2,1	2	3,1	5
0,2	1		0,2	5	1,2	4	2,2	5	3,2	8
0,3	0		0,3	10	1,3	9	2,3	10	3,3	13

Table 7. Calculation of SEDs for t = 6

2,0	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
2,0	0	2,3	0,0	13	1,0	10	2,0	9	3,0	10
2,1	1		0,1	8	1,1	5	2,1	4	3,1	5
2,2	4		0,2	5	1,2	2	2,2	1	3,2	2
2,3	9		0,3	4	1,3	1	2,3	0	3,3	1

Table 8. Calculation of SEDs for t = 7

1,3	SED	r_t	S_0	SED	S_1	SED	S_2	SED	S_3	SED
1,0	9	3,1	0,0	10	1,0	5	2,0	2	3,0	1
1,1	4		0,1	9	1,1	4	2,1	1	3,1	0
1,2	1		0,2	10	1,2	5	2,2	2	3,2	1
1,3	0		0,3	13	1,3	8	2,3	5	3,3	4

Output = {0, 0, 2, 1, 3, 0, 2, 1,}

We select the device XC3S400 which contains 400k gates, family is Xilinx Spartan-3, package PQ208, and speed -5 are used to implement the hardware for STTC encoder and decoder. The Table 1, Table 2 and Table 3 describe the hardware utilization for encoder and decoder respectively.

We note that the maximum space utilization for an encoder in a device available space is just 10%. The five 1-bit registers, two 3-bit registers, two 2-bit Latches, nine 1-bit xors are used. The minimum period: 2.882ns (Maximum Frequency: 346.933MHz), minimum input arrival time before clock: 4.161ns, and maximum output required time after clock: 6.141ns for device speed grade: -5 is used.

We note that the maximum space utilization for a decoder in a device available space is just 22%. The one 4x1-bit ROM, two 4x2-bit ROMs, one each 4x6-bit ROM and 8x6-bit ROM, twelve 7x7-bit multipliers, sixteen 6-bit adders, fifteen 7-bit adders, three 7-bit subtractors, three 1-bit registers, two 2-bit registers, eight 6-bit registers, sixteen 6-bit latches, forty one 6-bit comparator equal, sixteen 6-bit comparator great equal, forty eight 6-bit comparator less, sixty four 6-bit comparator less equal, eighteen 8-bit comparator less equal, and seventeen 6-bit 4-to-1 multiplexers are used. The minimum period: 12.584ns (Maximum Frequency: 79.466MHz), minimum input arrival time before clock: 3.442ns, and maximum output required time after clock: 6.216ns for a device speed grade: -5 is used. The trellis structure for received symbols is shown in Figure 5. The survive paths are shown in bold lines in Figure 7.

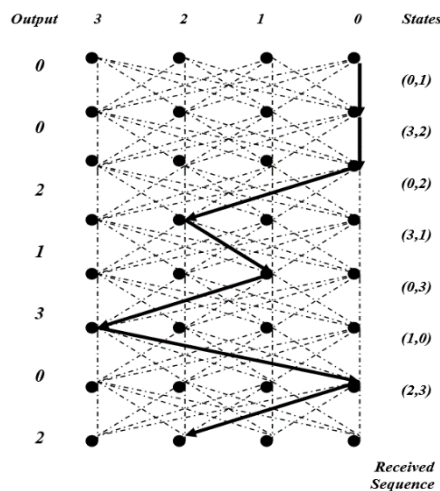


Figure 7. Trellis diagram for 4-state STTC decoder

The device logic utilization like number of slices, number of slice flip-flops, number of four input LUTs, number of bonded IOBs, number of multipliers, number of GCLKs for STTC encoder, decoder and STTC system are shown in Table 9, Table 10 and Table 11 respectively.

Table 9. Device utilization for encoder

Logic Utilization	Used	Available	Utilization
Number of Slices	9	3584	0%
Number of Slice Flip Flops	10	7168	0%
Number of 4 input LUTs	14	7168	0%
Number of bonded IOBs	10	141	7%
Number of GCLKs	1	8	12%

Table 10. Device utilization for decoder

Logic Utilization	Used	Available	Utilization
Number of Slices	718	3584	20%
Number of Slice Flip Flops	125	7168	1%
Number of 4 input LUTs	1304	7168	18%
Number of bonded IOBs	7	141	4%
Number of MULT18X18s	10	16	62%
Number of GCLKs	2	8	25%

Table 11. Device utilization for STTC system

Logic Utilization	Used	Available	Utilization
Number of Slices	720	3584	20%
Number of Slice Flip Flops	138	7168	1%
Number of 4 input LUTs	1307	7168	18%
Number of bonded IOBs	5	141	3%
Number of MULT18X18s	10	16	62%
Number of GCLKs	2	8	25%

4. SIMULATION RESULTS

The simulation results for STTC encoder and decode are shown below. We select the device XC3S400 which contains 400k gates, family is Xilinx Spartan-3, package PQ208, and speed -5 are used to implement the hardware for STTC encoder and decoder. The Table 1 and Table 2 describe the device/hardware utilization for encoder and decoder respectively. We note that the maximum space utilization for an encoder in a device available space is just 10%. These details are given below:

The five 1-bit registers, two 3-bit registers, two 2-bit Latches, nine 1-bit xors are used. The minimum period: 2.882ns (Maximum Frequency: 346.933MHz), minimum input arrival time before clock: 4.161ns, and maximum output required time after clock: 6.141ns for device speed grade: -5 is used.

We note that the maximum space utilization for a decoder in a device available space is just 22%. These details are given: The one 4x1-bit ROM, two 4x2-bit ROMs, one each 4x6-bit ROM and 8x6-bit ROM, twelve 7x7-bit multipliers, sixteen 6-bit adders, fifteen 7-bit adders, three 7-bit subtractors, three 1-bit registers, two 2-bit registers, eight 6-bit registers, sixteen 6-bit latches, forty one 6-bit comparator equal, sixteen 6-bit comparator great equal, forty eight 6-bit comparator less, sixty four 6-bit comparator less equal, eighteen 8-bit comparator less equal, and seventeen 6-bit 4-to-1 multiplexers are used. The minimum period: 12.584ns (Maximum Frequency: 79.466MHz), minimum input arrival time before clock: 3.442ns, and maximum output required time after clock: 6.216ns for a device speed grade: -5 is used. The simulation results for STTC encoder and decoder are shown in Figure 8, Figure 9, and Figure 10.

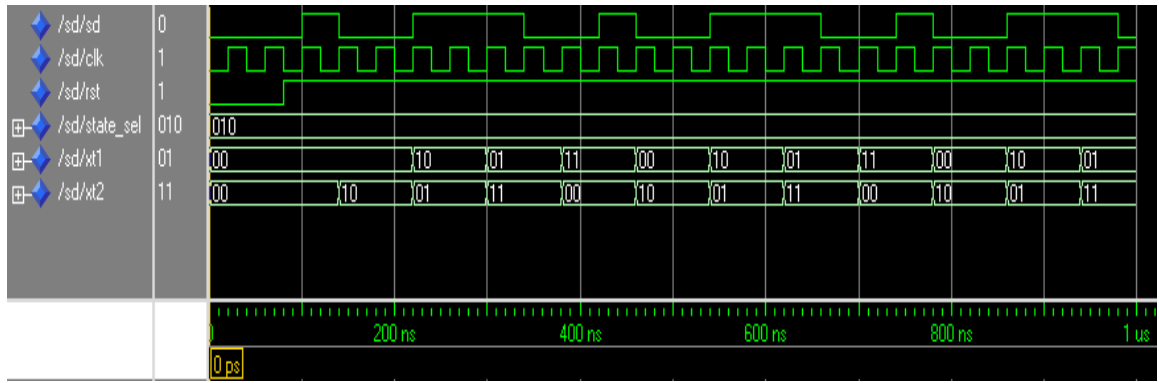


Figure 8. Output of STTC encoder

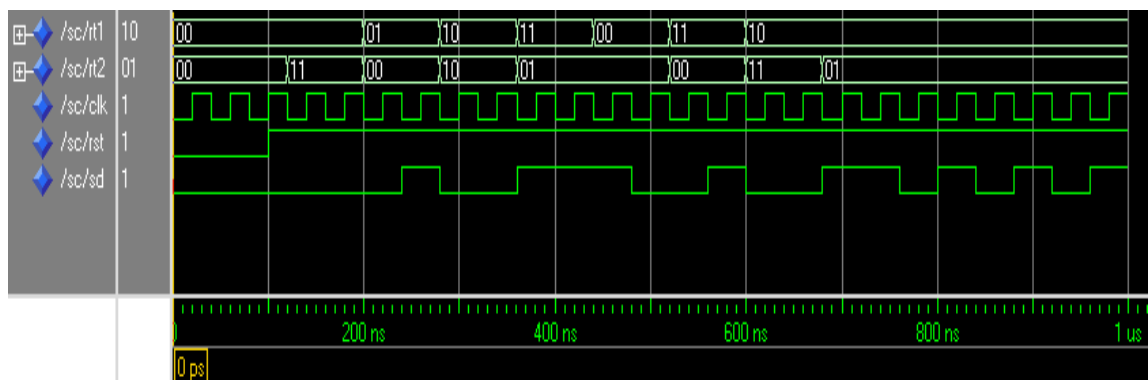


Figure 9. Output of STTC decoder with an error

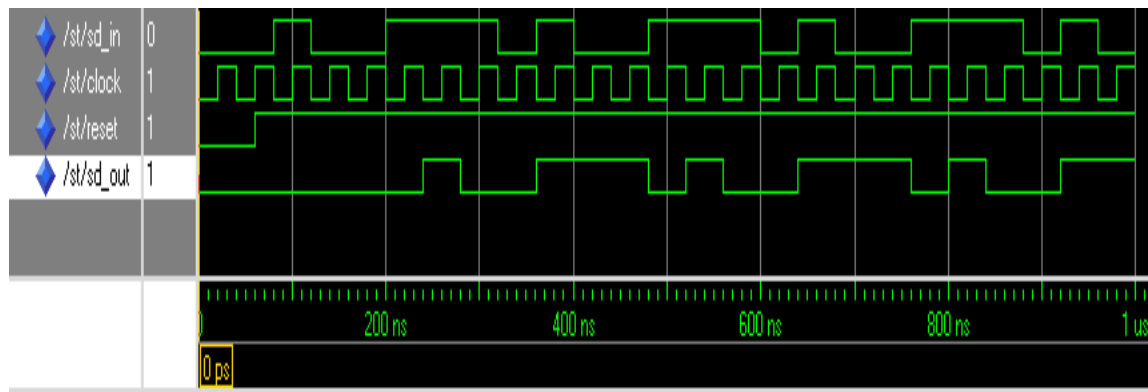


Figure 10. Output of both STTC encoder and decoder output

5. CONCLUSION

In this paper we have presented the implementation of STTC encoder and decoder for 4-state on FPGA. The device XC3S400 which contains 400k gates, family is Xilinx Spartan-3, package PQ208, and speed -5 are used for STTC implementation.

In STTC decoder implementation, the decoding complexity increases with number of states, therefore, an optimal decoding technique such as soft decision Viterbi decoding algorithm is used and makes it easy hardware implementation, reduces the complexity in branch metrics calculations, and hardware cost. However, the branch metrics calculations and difficulties in detector significantly increase with number of states. The logic utilization for STTC encoder and decoder is 10% and 22% as that of available device resources, respectively.

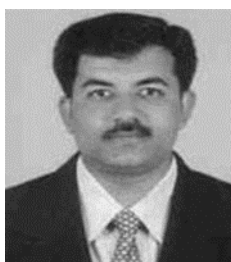
REFERENCES

- [1] A. Slaney and Y. Sun "Space-time coding for wireless communications an overview," *IEEE Proc.-Commun.*, vol. 153, no. 4, pp. 509-518, 2006.
- [2] S. M. Alamouti, "A Simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1451-1458, 1998.
- [3] V. Tarokh, N. Seshadri and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744-765, 1998.
- [4] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, 1996.
- [5] H. Suzuki, "A statistical model for urban multipath propagation," *IEEE Transaction on Vehicular Technology*, vol. 25, pp.673-680, 1977.
- [6] H. Hashemi, "Simulation of the urban radio propagation channel," *IEEE Transaction on Vehicular Technology*, vol. 28, pp.213-225, 1979.
- [7] Sundberg, C.-E. W. and Seshadri, N., "Digital cellular systems for North America," *IEEE Globecom*, pp. 533-7, Dec. 1990.
- [8] Balaban, N. and Salz, J., "Dual diversity combining and equalization in digital cellular mobile radio," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 342-54, May 1994.
- [9] Winters, J., Salz, J. and Gitlin, R. D., "The impact of antenna diversity on the capacity of wireless communication systems," *IEEE Trans. on Communications*, vol. 42, pp. 1740-51, Apr. 1994.
- [10] C. Zhipei, W. Zhongfeng and K.K. Parhi, "Iterative decoding of space-time trellis codes and related implementation issues," *Conjrence Record ofthe Thiry-Fotllzh Asihmar Conj Signah, Sys. Con@*, vol. 1, pp. 562-566, 2000.

BIOGRAPHIES OF AUTHORS



Mallikarjuna Gowda C.P received B.E., in Electronics and Communication Engineering from Gulbarga University, Karnataka, and M.E., in Electronics and Telecommunication Engineering from Jadavpur University, Kolkata, India in 1997 and 2002 respectively. He is currently pursuing his Ph.D. in Visvesvaraya Technological University, Karnataka, India. He is working as an Associate Professor in the department of Electronics and Telecommunication Engineering at BMS Institute of Technology and Management, Bengaluru, since February 2004. He has published several research articles in national and international journals/conferences, reviewer for IEEE Access journal. His research interests include Cognitive radio networks and its Applications, Resource allocation in CRN.



Dr. Raju Hajare received B.E degree in Electronics and Communication Engineering from Mysore University, India. He obtained his M. Tech in the field of Power Electronics from Visvesvaraya Technological University in 2006. He has been awarded with Ph.D. from VTU, Belagavi for his research Thesis on nano devices modelling and characterization. He is currently working as an Associate Professor in Electronics and Telecommunication Department of BMS Institute of Technology, Bangalore. He has won three best papers awards in international conferences. He has participated as member of panel of Judges in Southern India Science Fairs 2017, 2018, 2019 and 2020 for evaluating Electronics projects. His areas of interest are Semiconductor Devices, Nano electronics and NEMS. He has Published research papers in Scopus indexed international Journals and presented papers at different international Conferences. He has more than 20 quality publications to his credit. Project group under his guidance received best project award from VTU and also received fund from ocean Engineering department of IISc, Bangalore in 2018. Dr. Raju Hajare has authored two text books published in Cengage Publications for Electronics and communication engineering UG programs.