# Reconfigurable Enhanced Path Metric Updater Unit for Space Time Trellis Code Viterbi Decoder

Mohd Azlan Abu[1,2], Harlisya Harun[3], Mohammad Yazdi Harmin[1], Noor Izzri Abdul Wahab[4],
Mohd Khairil Rahmat[5], Mohd Fairuz Abd Hamid[5], Aizat Faiz Ramli[2]
*[1]Department of Aerospace Engineering, Faculty of Engineering,*
*Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
*[2]Universiti Kuala Lumpur British Malaysian Institute, 53100 Gombak, Selangor*
*[3]Universiti Kuala Lumpur Malaysia Institute of Aviation Technology, 43900 Dengkil, Selangor*
*[4]Department of Electrical and Electronic Engineering, Faculty of Engineering,*
*Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
*[5]Universiti Kuala Lumpur International College, 53100 Gombak, Selangor*
*mohdazlan@unikl.edu.my*

***Abstract*—Space Time Trellis Code (STTC) encoding and decoding techniques are effective for delivery of a reliable information because of the signal to noise ratio is very small. Even though the Viterbi algorithm is complicated to be designed, these methods typically used large memory space to store the information that have been processed mainly at the Path Metric Updater (PMU). Therefore, an effective memory management technique is one of the key factors in designing the STTC Viterbi decoder for low power consumption applications. This paper proposed the PMU memory reduction technique especially on Traceback activities that usually required a lot of memories for storing the data that has been processed in the past part by using Altera Quartus 2 and 0.18 µm Altera CPLD 5M570ZF256C5 as targeted hardware. Through this method, the reduction achieved at least 66% of memory requirements and 75% improvements in processing time without a significant effects on the outputs results of the STTC Viterbi Decoder for 4-PSK modulation technique by using 50MHz clocks.**

***Index Terms*—Space Time Trellis Code; Viterbi Decoder; Register Transfer Level.**

## I. Introduction

Wireless technology that has many transmit antennas and receive antennas for sending and receiving multiple data at the same time is called Multiple Input Multiple Output (MIMO) technique. Most electronic devices with wireless 802.11n standard support the MIMO, which is part of the technology that enables 802.11n equipment to achieve transmission speeds and higher revenue from the product without the 802.11n standard. MIMO technology is used either by the user (mobile) or forwarding to the access point (AP). For optimum performance, both the receiver and transmitter must use MIMO technology. MIMO technology is known in applying radio waves through natural phenomenon that called multipath propagation. By using this technique, the information will be transmitted to the buildings, hills, multiple objects and will be received by the multiple receiving antenna through different location and at different times [1].

MIMO technology is divided into two main sections which are spatial multiplexing (SM) and diversity coding. One of the main advantages obtained through spatial multiplexing MIMO technique is this technique can provide additional data capacities. MIMO spatial multiplexing gains these benefits by using various routes and the technique is effective to be used as a "channel" for sending additional data. The disadvantage of this technique is its requires information channels at the transmitter. When there are no channel informations at the transmit antennas, the diversity techniques is better to be used. In the method of diversity techniques, the flow (unlike the spatial multiplexing technique) is sent, but the signal is encoded using a technique called Space Time Coding (STC). A space time code (STC) is an excellent method to enhance the performance of transmitted informations in wireless communication channel that uses a lot of the transmitting antennas [2].

STC's performance depends on the number of transmitters. Multiple copies of data with the same information flow to a recipient in the hope that minimum values of the data from the transmitter may be survived between transmitter and receiver in a state that is good enough to allow decoding is performed. STC can be split into two main categories which are Space Time Trellis Code (STTC) which distributes the code trellis over multiple antennas and multiple time slots and provides both coding and diversity gain. Space time block codes (STBC) acts on a block of information at once (the same as blocking codes) and also provides diversification benefits but does not provide coding gain. Therefore, STTC techniques have a better ability of the technique for recovering of severe deteriorations than the STBC MIMO technique. However, in adopting the STTC Viterbi decoding algorithm which requires the processing of decoding complexity greater than STBC decoder. Therefore this technique is rarely used as STTC encoding techniques in wireless communication systems [2].

In previous publications, the Viterbi decoder Intergrated Circuit (IC) was designed using a convolution code (CC) and trellis-coded modulation (TCM). Some architectural Viterbi [2] - [8] had been proposed. This design used a memory unit such as Survival Metric Unit (SMU) to produce the desired output. On the other hand, [9] was designed the Viterbi Decoder without memories that improved the low-power design.

In this paper [9], the design and implementation in remodeling various low power technique which used the Viterbi decoder was presented. This design used a modified lists exchange strategy for developing memoryless Viterbi

Decoder. Through this method, based on a list of indicators of exchange method combined with a systolic design, this caused a reduction in the critical path of all systems compared to the previous method and system eventually led to the faster processing and higher speed. Design reworked and time multiplexing are the methods explored in order to create a complete system. This paper contributed to savings on the cost of processing time. This design was simulated using Verilog software and hardware implemented in Xilinx Virtex II pro xc2vp307fg676 by using system generator tool.

In this paper [10], a novel design memoryless Viterbi decoder for LTE system with 4-level soft decision was presented. Based on the proposed Viterbi Decoder designed, SMU part can be removed because the SMU used almost 50% of the power consumption. The proposed design was able to get a reduction of lower power compared to the existing one. In addition, this method can precisely process the output without using the SMU, it had a small amount of the processing time that had two clock cycles. This value is also much lower than existing Viterbi technique.

In the paper [11], the architecture of Viterbi Decoder was proposed consisting of branch units configure metrics (RBMU), memory metric way (PMM), a unit add compare select (CSU), a path selection unit (PSU) and the path tracer unit (PTU). The comparison with the previous hardware Viterbi Decoder and the proposed design that does not required a surviving memory unit.

However, the design with STTC decoder rarely was used or implemented as a complicated processing technique. To resolve this issue, the paper that proposed Path Metric Updater (PMU) without Traceback Unit was proposed. Then, the STTC Viterbi decoder was designed and implemented using 0.18 µm technology. In the current literature reviews, this is the first STTC Viterbi Decoder that has been designed without Traceback Unit is proposed and implemented.

This paper is organised as follows. Algorithms and the design of the existing STTC are described in section 2. The proposed Path Metric Updater for STTC decoding algorithm has been introduced in section 3. Results and discussion are presented and discussed in section 4. Finally, the conclusions for this works is provided in section 5.

## II. SPACE TIME TRELLIS CODE

The Space Time Trellis (STTC) MIMO system is shown in Figure 1 contains the STTC encoder with signal modulation processing, the Viterbi decoder for receiving signal correction code and signal diversity techniques. At time t, two bits of data which are $C_{t1}$ and $C_{t2}$ are encoded in the transmitter. Then, the STTC encoded symbols $X_t$, which are $X_{t1}$, $X_{t2}$ and $X_{t3}$ are transmitted by three transmit antennas. In the receiver parts, three NRx receiver antennas received the signals Rt which are Rt1 and Rt2 and decoded by the STTC Viterbi decoder and the output bits will be produced.

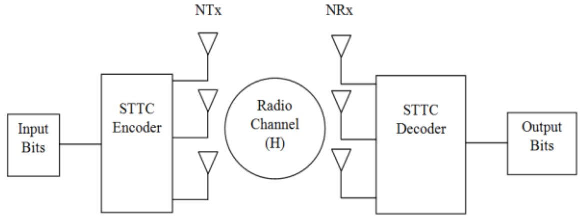

Figure 1: STTC MIMO diagram [1]

### A. STTC ENCODER

For the 4-PSK transmission modulation techniques, the STTC encoder is transmitted four inputs bit signal which are $C_t^1$ and $C_t^2$. Each input bit signals are sent to two delay's components $V_k$ which are $C_{t-1}^1$ and $C_{t-1}^2$ and are being encoded by using the set of generator coefficients $g_{j,i}^k$ comparable with the STTC code for the i number of transmit antenna, where $g_{j,i}^k$ =($g_{0,i}^k$ , $g_{1,i}^k$ ,.., $g_{Vk,i}^k$) for the k number of bit sequences and k = 1,2….,m. The encoded data $X_t^i$ are shown by Equation (1). The generator coefficients are accordingly designed using Equation (1).

$$X_t^i = \sum_{k=1}^{m}\sum_{j=0}^{v_k} g_{j,i}^k \; C_{t-j}^k \mod 2^m, \quad i = 1,2, \dots\dots, n_t \tag{1}$$

One way to show the STTC encoder is with the states diagram as shown in Figure 2. The state diagram shows the possible contents of the 4 states of the memories and the paths between the states perform the input bits and output bits resulting from each state transitions. The states of the register are declared as S0= 00, S1=01, S2= 10, S3= 11. The states diagram in Figure 2 shows all the state transitions from previous state to the next state that are possible for the STTC encoder. There are four inputs and 4 outputs from each state.

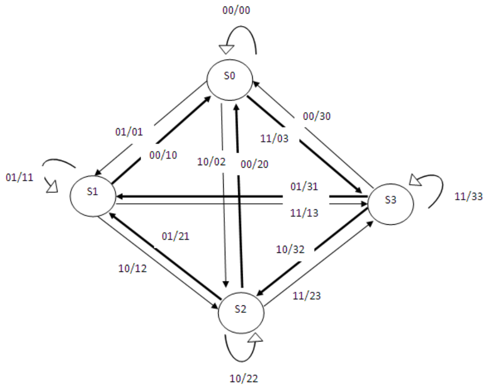

Figure 2: STTC Encoder diagram

### B. STTC VITERBI DECODER

To decode the received multiple signals that are transmitted using the MIMO medium, the STTC Viterbi decoder has been used by using the Viterbi algorithm as an error correction code. The STTC trellis refers to the modulation technique in the 4-PSK states diagram and there are four transition routes from the previous to the next states of 4-PSK. The modulation technique gives a huge impact on the complexity of the calculations on the branch and metric computation (BMC), add-compare-and-select (ACS) computation and Path metric updater computation (PMU) as shown in Figure 3.

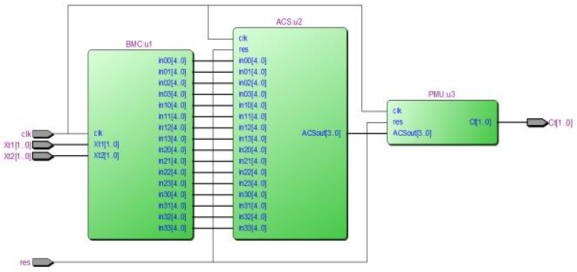

Figure 3: STTC Block Diagram

In an STTC Viterbi decoder, the values of the BMC are calculated using the Euclidean distance between the faded candidate symbol and actually received symbol for 4 states Trellis movement as shown in Equation (2) to Equation (5). Where $r_t$ is the received symbol in the receiver at time t. $h_t$is the channel effects between the receiver part and the transmitter part at time t, and $X_t^i$ is the candidate symbols from the transmitter part at time t. The branch metric calculations for each trellis lines are calculated and compared to retrieve the excellent received candidate symbol. It shows that the branch metric computation complexity is increased along with the number of receiver and the number of states.

$$BM(0,0) = |r_t - \sum h_t\, x_t^{(0,0)}|^2 \quad (2)$$

$$BM(1,0) = |r_t - \sum h_t\, x_t^{(1,0)}|^2 \quad (3)$$

$$BM(2,0) = \left|r_t - \sum h_t\, x_t^{(2,0)}\right|^2 \quad (4)$$

$$BM(3,0) = |r_t - \sum h_t\, x_t^{(3,0)}|^2 \quad (5)$$

The Add, Compare and Select function are comparing the Branch Metric Values for each path and choose the optimum value which is written and stored in the ACS outputs memory. The calculation for each state is calculated using the Equation (6) to Equation (9).

$$s0_n = \min(s0_{n-1}, b_{10} + s1_{n-1}, b_{20} + s2_{n-1}, b_{30} + s3_{n-1}, b_{30}) \quad (6)$$

$$s1_n = \min(s0_{n-1}, b_{11} + s1_{n-1}, b_{21} + s2_{n-1}, b_{31} + s3_{n-1}, b_{31}) \quad (7)$$

$$s2_n = \min(s0_{n-1}, b_{12} + s1_{n-1}, b_{22} + s2_{n-1}, b_{32} + s3_{n-1}, b_{32}) \quad (8)$$

$$s3_n = \min(s0_{n-1}, b_{13} + s1_{n-1}, b_{23} + s2_{n-1}, b_{33} + s3_{n-1}, b_{33}) \quad (9)$$

The path metric updater (PMU) architecture shows in Figure 4 to Figure 6 contain twelve logic elements. For the path metric updater (PMU) part after the Add, compare and its optimum path value is calculated in ACS, memories are written by using k=7 as shown in Figure 5 and the last computation is decoded by using the trace back algorithm [7]. Based on these figures, the outputs from ACSout are written in memories register for each path. Then the traceback algorithm is started to read the values in seven registers and the values are converted to the final results based on the look-up table. Figure 6 shows the path metric updater for look-up table based on the register transfer level diagrams. The inputs data from Figure 6 are taken from Figure 5 which are the memory values in seven registers. The three multiplexer will select the output results based on the input bits in the memories. For multiplexer number 1 (Mux 1) the output bits is 2’b01, for multiplexer number 2 (Mux 2) the output bits is 2’b10, for multiplexer 3 (Mux 3) the output bits is 2’b11. For others inputs, the output results is 2’b00.

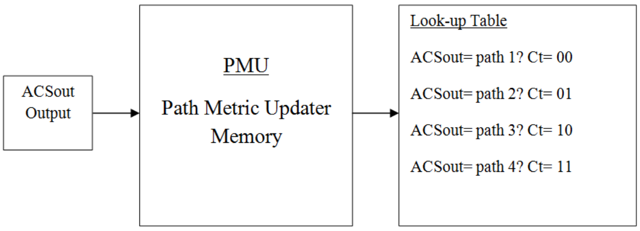

Figure 4: Path metric updater block diagram

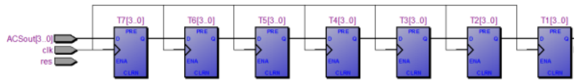

Figure 5: Path metric updater memory RTL

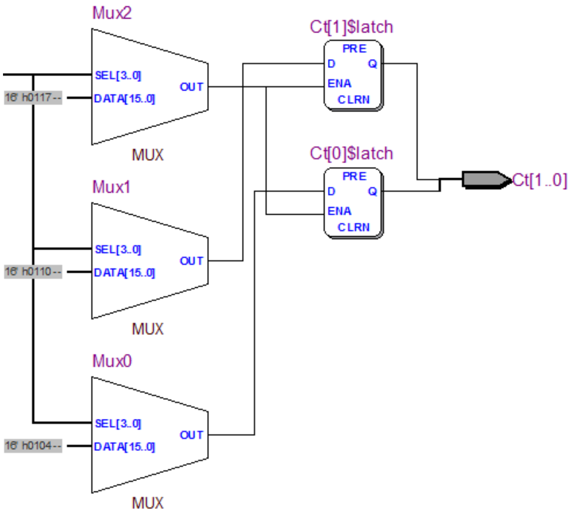

Figure 6: Path metric updater Look-up Table RTL

## III. Proposed Path Metric Updater Method

The proposed Path metric Updater is shown in Figure 7. The algorithm is declared as module Path Metric Up that contains ACS output results, 50 MHz clock cycles for synchronization, reset input for circuit reset to state 0 and Co as PMU output of this module. The parameter such as clock, reset and ACS are declared as input signals and the parameter Co is declared as an output signal. The output signal needs a register to store the last computation value in PMU. The *always* loop is declared and the input signal is taken from the ACS output and declared as Input Bit 1 to Input Bit 5. The output signals are set based on the look-up table same as the previous section. Inside the *always* loop, there is the *case* statement to map the input signal with the output signals. This *case* statement needs to be closed by using the *endcase* statement. Then the *always* module needs to be closed loop by using *end* statement. Lastly after all the computations in this algorithm are completed, the module needs to be closed by using the *endmodule* statement.

```
module Path Metric Up (ACS, clock, reset, Co);
input clock, reset;
input [3:0] ACS;
output [1:0] Co;
reg [1:0] Co;
always @(Table) begin
        case (Table)
              Input Bit 1 : Output Bit = 2'b00;
              Input Bit 2 : Output Bit = 2'b01;
              Input Bit 3 : Output Bit = 2'b10;
              Input Bit 4 : Output Bit = 2'b11;
              Input Bit 5 : Output Bit = 2'b00;
        endcase
end
endmodule
```

Figure 7: Proposed PMU Algorithm

## IV. Results and Discussions

The improved path metric updater (PMU) register transfer level (RTL) contains one multiplexer, two OR gates and one register to store the final PMU results as shown in Figure 8. The function of the multiplexer is to decode the trace back results to the correct values based on the decoding look-up table that has been explained in part three before. The registers from the existing design and the process of trace back have been removed. Total logic element for this design is four logic elements.

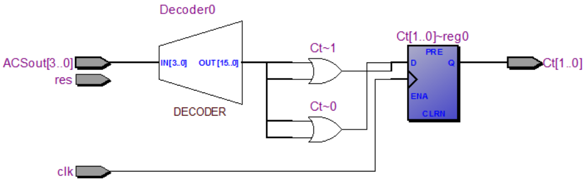

Figure 8: Path metric updater Look-up Table RTL

Figure 9 shows the results for the existing PMU in 5 clock cycle. The ACSoutput is the input of the PMU block circuit. The ACSout are set as a 0000, 0001, 0010, 0100 and 1000 to represent the input signals. The ACSout [3] to ACSout [0] are representing the bit values in ACSout. T1 to T7 represent the memories value for k=7. The clock signal is set as a positive triggered clock. The $C_t$ is the output signal from the PMU.

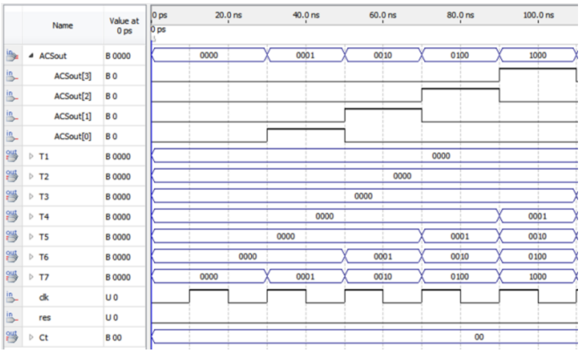

Figure 9: Existing Path metric updater For 5 Clocks Cycle

The output signal in PMU, $C_t$ has produced the results after seven clock cycle. In 8th clock cycle the $C_t$ is 00, in 9th clock cycle the $C_t$ is 01, in 10th clock cycle the $C_t$ is 10 and in11th clock cycle the $C_t$ is 11 as shown in Figure 10.

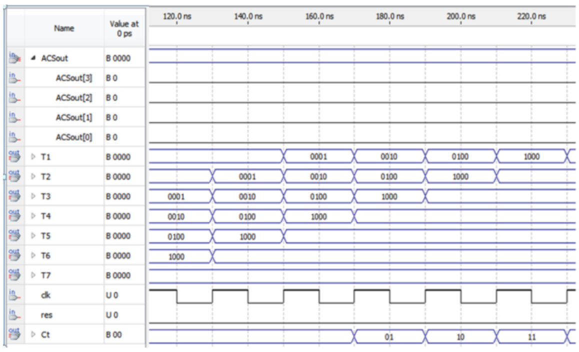

Figure 10: Existing Path metric updater For 5 Clocks Cycle

Figure 11 shows the improved PMU results before three clock cycles. It is shown that the output is produced as $C_t$ at clock number 2 which the output bit is 00.

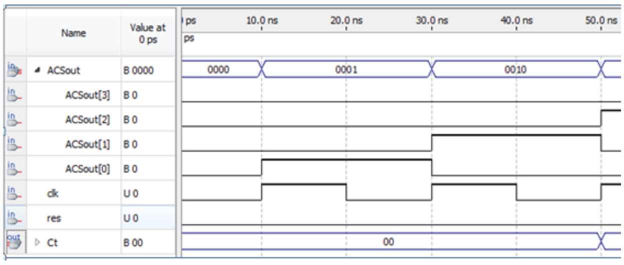

Figure 11: Improved Path metric updater before 3 Clocks Cycle

Figure 11 shows the improved PMU results after three clock cycles. It is shown that the output is produced as $C_t$ at clock 3 which the output bit is 01. At clock 4 the $C_t$ produced the output of 10 and at clock 5 the $C_t$ produced the output of 11.

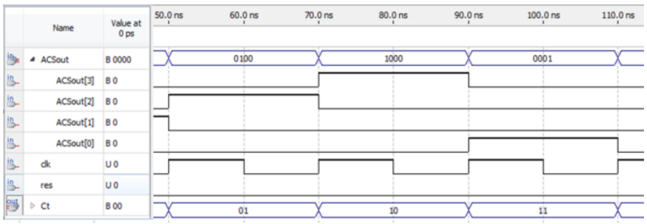

Figure 12: Improved Path metric updater after 3 Clocks Cycle

The results for improved PMU is similar for the inputs and outputs bits but the performance is better in terms of outputs processing time because the improved PMU can produce the outputs after two clock cycle compared to existing PMU that needs eight clock cycles to produce the outputs.

## V. Conclusion

In this paper, an improved PMU and its Register Transfer Level (RTL) hardware architecture are proposed and tested by using Altera Quartus 2 software and implemented in the 0.18 µm Altera CPLD 5M570ZF256C5 as the targeted hardware. This proposed design has improved the logic element by 66% and 75% faster processing time compared to the existing design.

## References

[1] K. T. Shr, H. D. Chen and Y. H. Huang, "A Low-Complexity Viterbi Decoder for Space-Time Trellis Codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, (2010), 873-885.

[2] D. A. El-Dib and M. I. Elmasry, "Modified register-exchange viterbi decoder for low-power wireless communications," *IEEE Trans. on Circuits and Systems I*, vol. 51, no. 2, (2004), 371-378.

[3] E. Boutillon and N. Demassieux, "High speed low power architecture for memory management in Viterbi decoder," *IEEE Int. Symp. Circuits and Systems*, vol. 4, (1996), 284-287.

[4] J. Ryu, S. Kim, J. Cho, H. Park, and Y. Chang, "Lower power Viterbi decoder architecture with a new clock-gating traceback unit," *6th Int. Conf. VLSI and CAD*, (1999), pp. 297-300.

[5] P. J. Black and T. H. Meng, "A 140Mbis 32-state radix-4 Viterbi decoder," *IEEE Trans. Solid State Circuits*, vol. 27. no. 12, (1992), 1877- 1885

[6] C. C. Lin, Y. H. Shih, H. C. Chang, C.Y. Lee, "Design of a power–reduction Viterbi Decoder of WLAN Application," *IEEE Trans. Circuit and Systems I*, vol. 52, (2005), 1148-1156.

[7] Yun-Ching Tang; Do-Chen Hu, Weiyi Wei, Wen-Chung Lin, and Hongchin Lin, "A memory-efficient architecture for low latency viterbi decoders," *IEEE Int. Symp. on VLSI Design, Automation and Test, VLSI-DAT '09*, (2009), 335-338.

[8] Rongchun Li, Yong Dou, Jie Zhou, and Guoqing Lei, “A High throughput Reconfigurable Viterbi Decoder,” *Int. Conference on Wireless Communications and Signal Processing*, (2011), 1-6.

[9] A. K. Mishra and P. P. Jiju, "Low power, dynamically reconfigurable, memoryless systolic array based architecture for Viterbi decoder," *Energy, Automation, and Signal (ICEAS), 2011 International Conference on Engineering and Applied Sciences*,(2011),1-5.

[10] C. Yu, Y. S. Su, B. S. Lin, P. H. Cheng and S. J. Chen, "A memoryless Viterbi decoder for LTE systems," *The 1st IEEE Global Conference on Consumer Electronics 2012*, (2012), 643-644.

[11] C. Yu, Y. S. Su, B. S. Lin, P. H. Cheng and S. J. Chen, "A dual-code-rate memoryless Viterbi decoder for wireless communication systems," *2013 IEEE International Conference on Consumer Electronics (ICCE)*, (2013), 578-579.

[12] M. Valinataj, “Reliability and Performance Evaluation of Fault-aware Routing Methods for Network-on-Chip Architectures”,.*International Journal of Engineering*, *27*(4 (A)), (2014), 509–516.

[13] M. Ahmadian-Attari, “Multilevel Input Ring-TCM Coding Scheme: A Method For Generating High-Rates Codes *International Journal of Engineering*, (2001), *14*(3), 209–214.

[14] S J Azhari and H Kaabi, “A Novel Data Compression Technique For 4-20 mA Current Loop Transmitters”, *International Journal of Engineering*, *14*(1), 35–40