Manchester encoder component for data processing IoT environment

Adi Putra Arianto¹, Sunny Arief Sudiro², Abdul Hakim² and Fivtatianti Hendajani²

¹Gunadarma University Department of Electrical Engineering
²STMIK Jakarta STI&K
aputraarianto028@gmail.com, sunnyarief@yahoo.com, hkiem09@gmail.com, fivtatantihoundajani@gmail.com

Abstract. Line encoding is crucial especially in data communication which part of internet of things environment. Line encoding doing encode raw bit stream and applying some rules before transmitting to transmission medium. Some example of encoding technique is: non-return zero encoding (NRZ), return-zero encoding, and Manchester encoding. Manchester coding technique have advantages compared to NRZ and RZ coding. Manchester coding can recognize bit interval and will have no problem to encode long same bit stream. Manchester encoder commonly built using logical XOR gate, with clock and bit stream as input and output of XOR gate will determined as encoded signal. The common design has a weakness since the clock is use in operation, and the output will be separate into two parallel bit stream. Common design also has spike glitch in some transitions. In this work, the new design of Manchester encoder using FPGA device is proposed to overcome common design problems and suitable to be implemented in IoT environment.

1. Introduction

The Internet of Things terms (IoT) first introduced by Kevin Ashton in 1999 to call a system of physical object that can connect to internet by sensor. IoT now become popular to describe internet connectivity and computing capability to connect to a variety objects of everyday items [1]

IoT is a network of physical objects. The internet is not only a network of computers, but it has evolved into a network of device of all type and sizes, vehicles, smart phones, home appliances, and others, all connected, all communicating & sharing information based on stipulated protocols in order to achieve smart reorganizations, positioning, tracing, safe & control & even personal real time online monitoring, online upgrade, process control & administration[2]

IoT do a lot of communication between device to device. Communication device to device used in IoT commonly use digital data and digital signal. Transmission system needs encoding for sending data or information through carrier signal in transmission medium.

Encoding is the process of translating bit stream into specified format in data communication. Encoding exist in physical layer of OSI layer. Several common techniques of encoding are non-return zero encoding, return-zero encoding, and Manchester encoding. Manchester coding technique have advantages compared to NRZ and RZ coding. Manchester coding can recognize bit interval and will have no problem to encode long same bit stream.
There is a research in [3] and [4], propose a method of building Manchester encoder using XOR gate operation between clock and signal data. XOR gate output is determined as encoded Manchester output. The design on the research has a drawback, the output of Manchester encoder determines as two parallel bit stream and causing output bandwidth requirement to twice of input bandwidth and needs two channels to send the encoded signal.

There is another research in [5] and [6] propose a same method of building Manchester encoder using XOR gate. In the research [5] Manchester encoder is used in UHF frequency which also has drawback that it will glitch when transition interbit and will not be efficient when implemented in UHF frequency.

This work is proposing new design of Manchester encoder with more suitable component instead of just XOR gate. This work will meet the requirement of Manchester encoder IEEE 802.4 truth table and only need one channel to send one-bit stream encoded signal. This work is organized as follow: Section 2 explains common Manchester encoding technique and the weakness. Section 3 explains the proposed design. Simulation and results of proposed design are described in section 4. Conclusion are in section 5. Design is built with VHDL in Xilinx 8.1, and simulation is done using Xilinx 8.1 ISE simulator.

2. Manchester encoding technique
There are two type of Manchester encoder, the first is by G.E Thomas in 1949, signal level of 0 is translated as transition from low to high, and signal level of 1 is translated as transition from high to low. The second type is followed by William Stallings and IEEE 802.4 says signal level of 1 is translated as transition from high to low, and signal level of 1 is translated as transition from low to high. Figure 1 shows the signal model of two type of Manchester encoder: [5]

![Figure 1. Type of Manchester Encoder][1].

Common Manchester encoding is realized with a XOR operation for clock and input signal [6]. The clock always has a transition within one cycle and so does the output no matter what the signal input is [3]. Figure 2 shows the Manchester encoder structure:

![Figure 2. Manchester encoder structure][2].

The example signal model of Manchester encoder when using input stream of 01101 is shown in figure 3:
Common Manchester encoding should meet the following: [4]

- If the input X is logic-0 and clock signal is logic-1, output signal will be determined as logic-1
- If the input X is logic-0 and clock signal is logic-0, output signal will be determined as logic-0
- If the input X is logic-1 and clock signal is logic-1, output signal will be determined as logic-0
- If the input X is logic-1 and clock signal is logic-0, output signal will be determined as logic-1

From that condition truth table can be generated as in Table 1:

| Input X | CLK  | Output |
|---------|------|--------|
| 0       | 1    | 1      |
| 0       | 0    | 0      |
| 1       | 1    | 0      |
| 1       | 0    | 1      |

The truth table of conventional Manchester Encoder is just identical as XOR gate truth table, that concluded Manchester encoder can be built using single XOR gate by using digital data and clock as input and will generate an output called as Manchester output.

3. Conventional and proposed design

We evaluate method based on theoretical information given, we build conventional Manchester encoder using Xilinx ISE 8.1. XOR gate is used to build Manchester encoder. VHDL language is used to generate the XOR gate. Time scale used on simulation are:

- Clock high time : 50ns
- Clock low time : 50ns
- Setup time : 1ns
- Output Valid Delay: 1ns
- Clock Edge : Rising Edge
Figure 4. Time scale used on Conventional Design simulations.

![Image](image_url)

The result by setting X input as 01101 is shown in figure 5:

![Image](image_url)

Figure 5. XOR Manchester simulation result.

From the signal simulation result, we found significant problem. Because clock is used to determine the output, output is force-triggered high in every rising edge of clock, hence make some spike glitch when the output should be low in clock rising edge state. This will cause decoder to falsely recognize the output, and cause lagging in higher frequency of stream.

We propose the design uses more complex logical gate instead of simple XOR gate. The design use JK flip-flop that use to divide clock to be able to synchronize in later operation. D flip-flop is used in the design as buffer of divider output to be an input of main Manchester encoder component. Figure 6 shows the RTL schematic of overall design.

![Image](image_url)

Figure 6. RTL schematic of Manchester encoder design.

The clk pin is use as main clock of operation, data pin is use as input signal, rst pin is use as reset, and dout pin is an output of Manchester encoder. Figure 7 shows the complete detail what is the component inside of Manchester encoder design.
Figure 7. RTL schematic of Manchester Encoder.

The clk pin is connected directly to CLK_IN. CLK_IN is the input of divider component which is JK flip-flop. CLK_OUT is the output of JK flip-flop. More detail of divider component shown in figure 8.

Figure 8. RTL schematic of Divider Component.

The reset pin is directly connected to reset pin of D flip flop and reset pin of main Manchester encoder. The function of reset pin is as trigger of reset in the component. Data pin is directly connected to d pin of D flip flop and will be use as input in D flip-flop. Figure 9 shows the detail of D flip-flop component.

Figure 9. RTL Schematic of D flip-flop Component.

The main Manchester encoder component use clock in CLK pin and use the signal from clk main clock of operation. RESET pin of this component use signal from main rst of operation. X pin is for input of unencoded data but already synchronized (frequency already divide by 2, output of divider component). Main Manchester encoder component output is directly connected to output pin “dout” and determined as encoded signal. Figure 10 shows the detail inside main Manchester encoder component.

Figure 10. RTL Schematic of Main Manchester Encoder Component.
4. **Simulation result**

We evaluate the new proposed design using Xilinx 8.1 ISE simulator. Time scale used on simulation are:

- Clock high time : 50ns
- Clock low time : 50ns
- Setup time : 1ns
- Output Valid Delay: 1ns
- Clock Edge : Rising Edge

The result of signal processed in divider component is shown in Figure 12a and Figure 12b:

![Figure 11. Time Scale Used in Proposed Design Simulation.](image)

The result of signal processed in divider component is shown in Figure 12a and Figure 12b:

![Figure 12a. Divider Component Signal.](image)

![Figure 12b. Divider Component Signal.](image)

Then the output of divider component used in D flip-flop component as input in enable pin. D pin of D flip-flop is connected to signal unencoded data. The processed signal in D flip-flop component is show in Figure 13a and Figure 13b:
Then the D flip-flop output is connected to pin x of main Manchester encoder and used as input signal. RESET pin of main Manchester encoder is directly connected to main rst pin. Clock used in main Manchester encoder is main clock. The result of signal processed in main Manchester encoder component shown in figure 14a and figure 14b:

From the figure 14a and figure 14b, we can conclude the truth table shown in table 2:

| Input Data | Output at 1st Half | Output at 2nd Half |
|------------|--------------------|--------------------|
| 0          | 1                  | 0                  |
| 1          | 0                  | 1                  |
From the table 2, we can compare the proposed design and conventional design:

- In conventional design (table 1) when $X = 0$, the output will be the transition between logic 1 to 0, with XOR operation between input 0 and CLK (rising edge). Output result is two parallel bit stream.
- In proposed design (table 2), when input $X = 0$, the output will be serially making logic 1 then 0 so output result will be just one-bit stream.
- In conventional design, when input $X = 1$, the output will be transition between logic 0 to 1, with XOR operation between input 1 and CLK (rising edge). Output result is two parallel bit stream.
- In conventional design, when input $X = 1$, the output will be serially making logic 0 then 1 so output result will be just one-bit stream.

The comparison between signal model of conventional Manchester and proposed Manchester design shown in figure 15a and figure 15b:

**Figure 15a.** Conventional Manchester Encoder Signal Result.

**Figure 15b.** Proposed Manchester Encoder Design Signal Result.

From the comparison we can conclude that conventional Manchester encoder needs two parallel bit stream as output and proposed design only need one of bit stream. From the signal model, it shows proposed design has no glitch and we can conclude that our proposed design is better than conventional Manchester encoder in terms of glitches. In terms of convention of Manchester encoder as per IEEE 802.4, it states that a logic 0 is represented by a high-low signal sequence and a logic 1 is represented by a low-high signal sequence, which in proposed design these rules is already fulfilled.

Resources needed to build the proposed design is shown as in figure 16:

**Figure 16.** Proposed Design Resources.
5. Conclusion

Internet of Things become crucial in human life these days. IoT needs device to device communication and digital data to digital signal communication system needs to be built to achieve term of Internet of Things. Digital data to digital signal communication needs encoding technique. Manchester encoder technique has more advantages than others, but conventional Manchester encoder has several drawbacks. In this work, VHDL language and Xilinx 8.1 is used to build Manchester encoder to reduce several drawbacks of conventional one. The proposed design successfully overcome the glitch that normally happen in conventional Manchester encoder. Proposed design also only needs one-bit stream as output instead of two. The encode speed of proposed design is half of main clock frequency, because the encode speed and main clock is different, the input signal frequency needs to be half of main clock to be able to synchronize encode speed of the design. The resource needed to build proposed design are 1 flip-flop, 2 combinational logic gate uses as latches, and 2 of four input LUTs. The resource needed in proposed design is more than needed in conventional one, because the conventional only need one XOR gate.

6. References

[1] Patel K and Patel S, 2016, Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges, in International Journal of Engineering Science and Computing, 6 Issue No. 5
[2] Rose.K, Eldridge.S, Chapin.L 2015, The Internet of Things: An Overview pp 12
[3] Lee Y.H and Pan C.W, 2015, Fully Reused VLSI Architecture ofFM0/Manchester Encoding Using SOLS Technique for DSRC Applications, in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 23 no. 1, pp. 18-29
[4] Lee Y.H, Pan C.W and Tsai F.H, 2012, VLSI Architecture Design of FM0/Manchester Codec with 100% Hardware Utilization Rate for DSRC-Based Sensor Nodes in ITS Applications, in Journal of Latex Class Files, 11, no 4
[5] Khan M.A, Sharma M, and Brahmaanadha P.R, 2008, FSM based Manchester encoder for UHF RFID tag emulator, in Proc. Int. Conf.Comput., Commun. Netw, pp1-6
[6] Prasad, Srinivas, Kiran and Varma, 2016, Design and Implementation of FM0/Manchester Encoder using VHDL, in International Journal of Advanced Research in Computer and Communication Engineering, 5, issue 4
[7] Stallings W, 2004, Data and Computer Communications (7th ed.), Prentice Hall. pp. 137–138. ISBN 0-13-100681-9