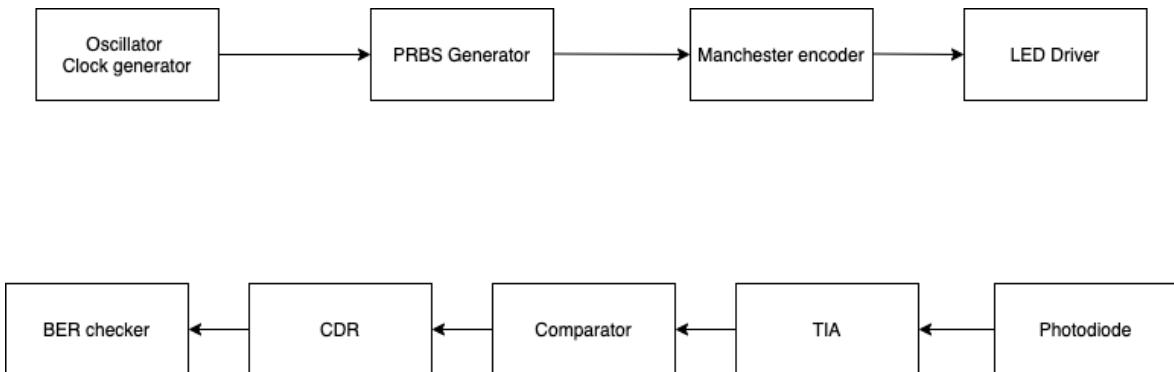


# Contents

1	System block diagram for synchronous communication	3
2	Clock generator / Oscillator	4
3	PRBS generator	5
4	Manchester encoder	6
5	Clock and Data Recovery(CDR)	9
5.1	CDR using one shot . . . . .	9
6	BER checker	11

# 1 System block diagram for synchronous communication

The following block diagram shows the various parts of circuit for synchronous underwater optical communication link. We aim to establish synchronous optical communication link. As we have one forward channel to transmit. We have to combine timing information with data itself and transmit. At the receiver side, we need a mechanism by which we can recover timing information from incoming data stream and then extract data from it. Manchester encoding scheme comes very helpful in this case. Hence we design circuit for transmitter and receiver based on manchester encoding and decoding as shown in Fig 1.1.



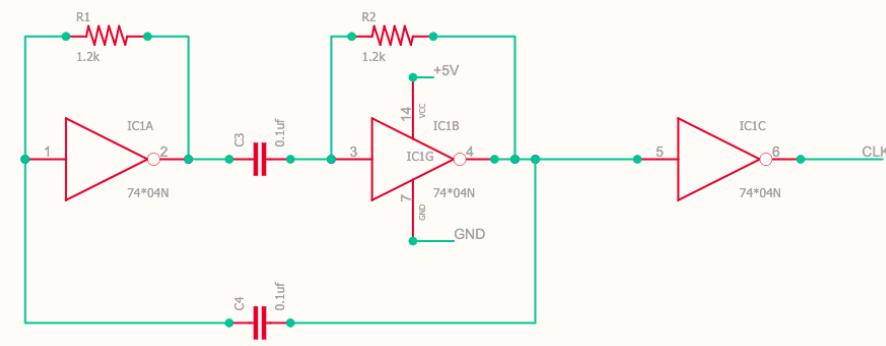
**Fig. 1.1.** System block diagram

As shown in Fig 1.1, upper block series is for transmitter. Initially we use simple clock generator or oscillator circuit to generate required clock. Various approaches are used to generate clock. These are discussed in section 2. Next block is PRBS generation where we generate random data to transmit. PRBS generator has auto start functionality as well. This is discussed in section 3. Next, data generated in PRBS block is encoded in Manchester format. In manchester encoding, the timing information or clock is embedded in data. Resultant signal is combination of clock and data. Now we can drive LED with this signal to transit data in optical format. This is explained in section 4.

At the receiver side, first we have photodiode followed by transimpedance amplifier which converts optical signal received into electrical signal. This has been discussed in section ???. Comparater block is used to threshold the signal and recover logic levels of signal. Then we have Clock and Data Recovery(CDR) block. CDR recovers clock from incoming data stream and extracts data from it. 2 methods implemented for CDR are explained in section 5. Once we recover clock and data at receiver side. We check for bit errors using BER checker block. We utilize the property of PRBS generator used at transmitter side to find out BER.

The complete block diagram is implemented in hardware using TTL IC's. Output of BER checker is error pulse. This means output of BER checker gives a pulse for each error in data. Lastly we measure BER by counting these pulses.

## 2 Clock generator / Oscillator

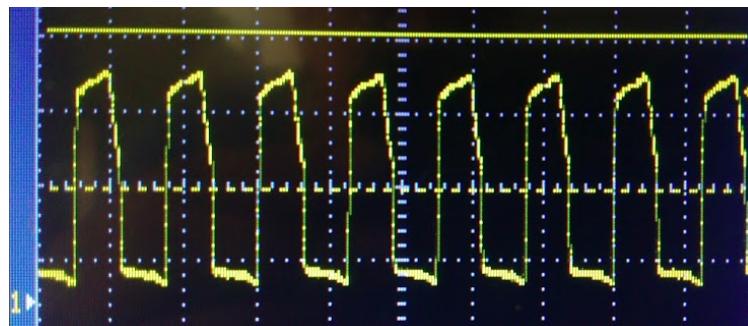


**Fig. 2.1.** Clock generator

Fig 2.1 shows fixed clock frequency generator. This circuit is based on 7404 gate. This is simple and low cost clock generator. For in-lab experiments we can use function generator which can give variable frequency. But for outdoor testing of communication link we need compact and on board clock generator. Hence we go for this simple fixed frequency oscillator. The clock frequency depends on value of resistor R1, R2 and capacitors C1, C2. Different values of these components are used and we get different frequencies as listed in following table.

R	C	Frequency
1.2k	0.1uF	10kHz
560	5nF	300kHz
560	1nF	1MHz

Output of this oscillator circuit is not perfectly stable frequency. Frequency may vary by some amount around central frequency. It was observed that signal generated after first two inverters is distorted and attenuated. Hence additional inverter was required as shown in the circuit. These two outputs are shown in following waveforms. Fig 2.2 shows generated square wave signal and Fig 2.3 shows buffered clock which determines rate of communication.



**Fig. 2.2.** Clock generated

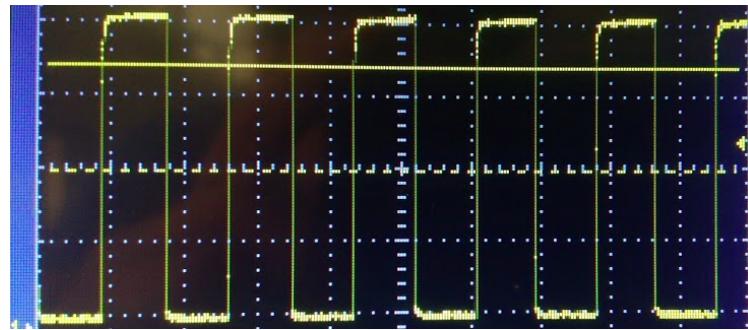


Fig. 2.3. Buffered clock

### 3 PRBS generator

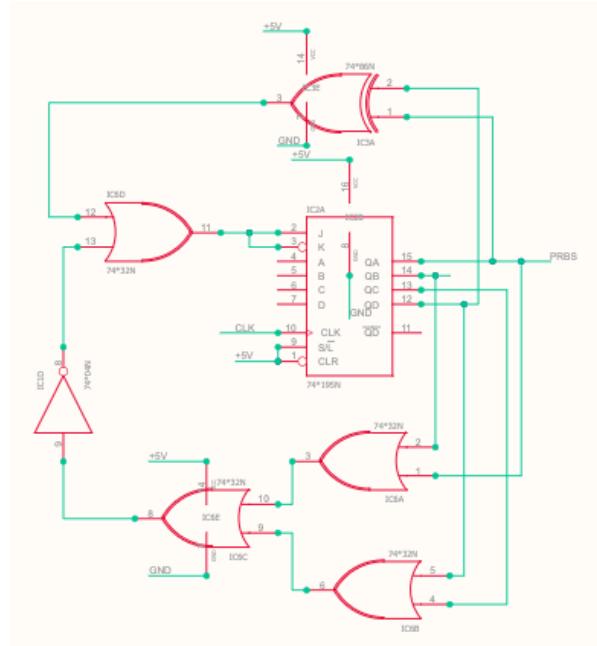
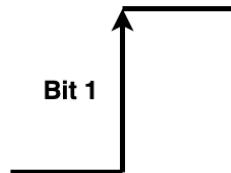


Fig. 3.1. PRBS generation with auto reset

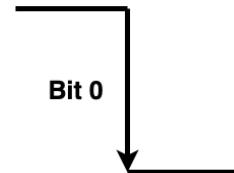
Above circuit generates PRBS data of repetition length 15. It is similar to one discussed in section ???. This PRBS generator has auto reset functionality. It may happen when power is on, all four bit positions in shift register are reset or logic 0. In this case Linear feedback shift register will not be able to generate PRBS data. Hence when such condition arises, we need to feed logic 1 into the shift register such that data in register becomes valid and it will start generating output again. We could have used a manual reset switch to feed logic 1 when such situation arises. But better approach can be to have a combinational circuit which can look for any occurrence of 0000 in the LFSR. When such condition occurs, logic 1 is gated through OR gate into  $Q_A$  position of shift register as shown in Fig 3.1. Hence along with Ex-OR feedback for PRBS generation, we implement auto-reset feedback also which has boolean expression as :  $(Q_A + Q_B + Q_C + Q_D)'$ . This feedback is gated using OR gate and fed to serial in input of 74195 shift register as shown in above circuit diagram.

## 4 Manchester encoder

Manchester encoding is a digital encoding where data bits are represented as transitions. In NRZ line encoding format we use stable logic levels. In manchester encoding data bits are mapped to logic transition from low to high and high to low. Following figure shows this mapping.

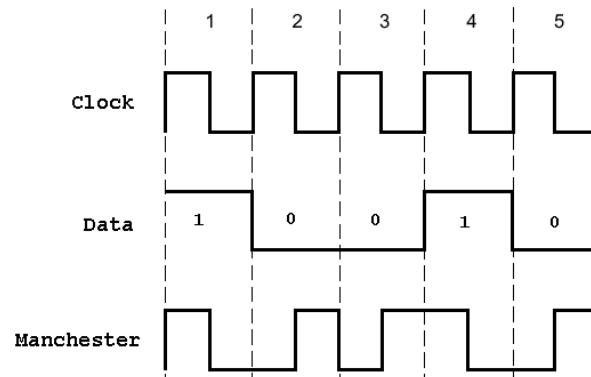


**Fig. 4.1.** Transition for Logic 1

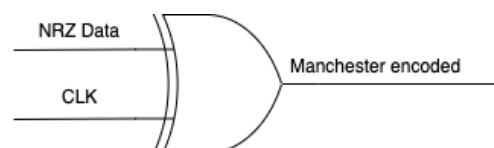


**Fig. 4.2.** Transition for logic 0

As shown in Fig 4.1 logic bit 1 is represented as transition from low to high. Logic 0 is represented as transition from high to low which is shown in Fig 4.2. In this way NRZ data is mapped to transitions. This is also called phase encoding. The advantage of doing this is self-synchronization. Manchester code has transition in the middle of each clock period. Hence it becomes easy for clock recovery. In case of NRZ, if we consider case of long strings of 1s or 0s, at the receiver we get no transitions and hence we loose synchronization. Hence no timing information is sent by NRZ data format. While in case of manchester, we get transition at each clock cycle. Hence we can recover clock and data from it using digital phase lock loop or using multiple monostable multivibrators. However the drawback of this method is that bandwidth of signal gets doubled as there are more transitions. Fig 4.3 shows concept of manchester encoding. We can note that there is transition at the middle of each clock cycle.



**Fig. 4.3.** Manchester encoding (source: matlab.com)

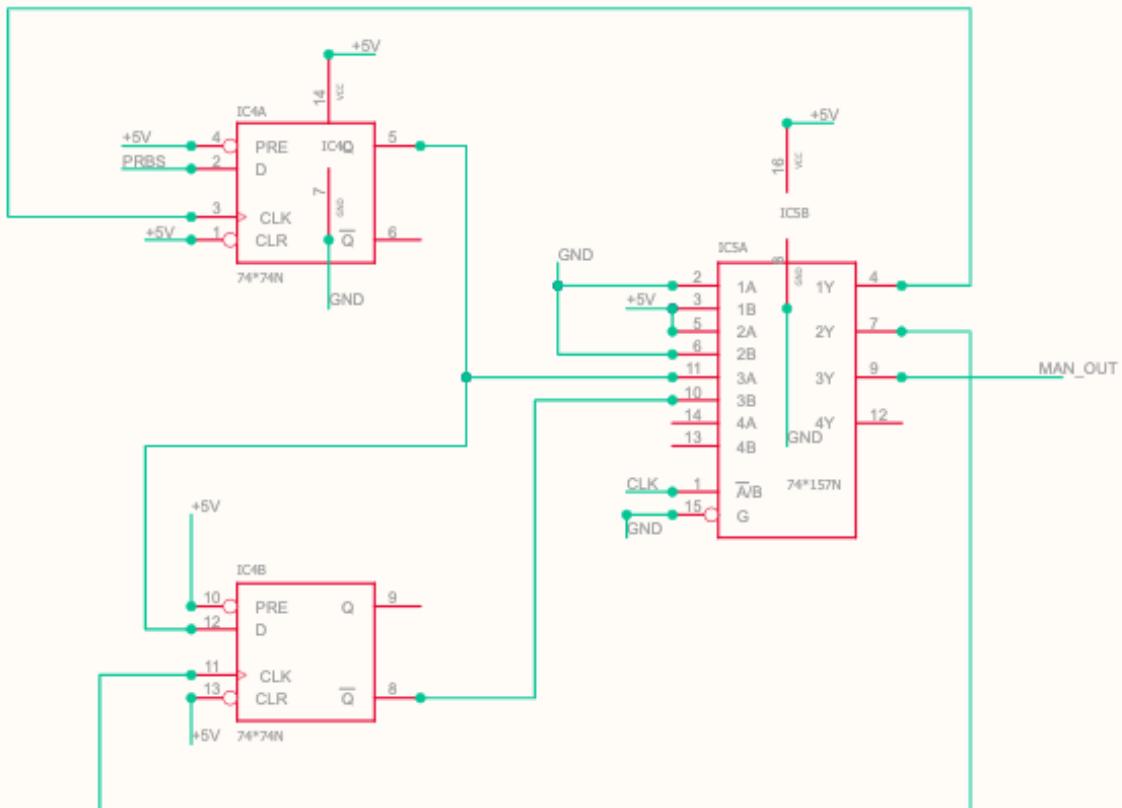


**Fig. 4.4.** Manchester encoding using ExOr

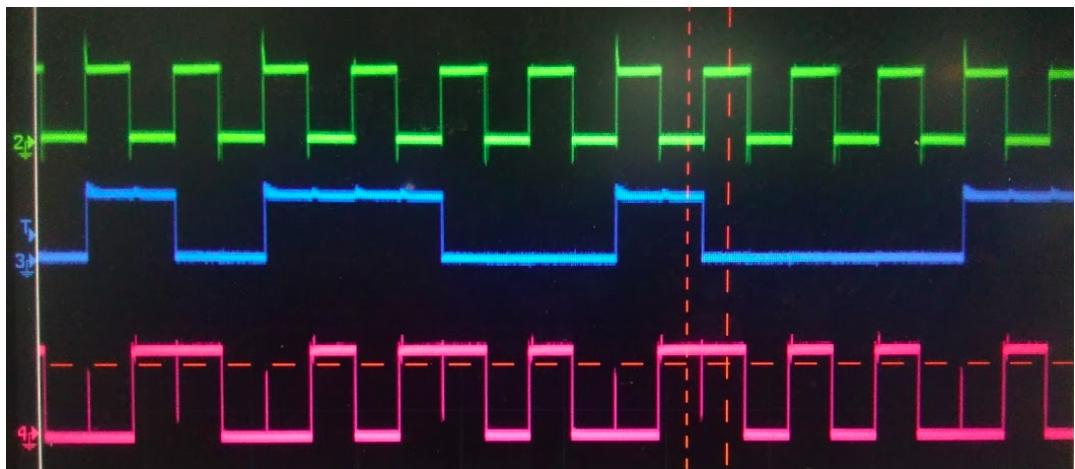
We can generate manchester encoded data just by ExRr logical operation between data and clock as shown in Fig 4.4. However simple ExRr may not work always as there might be slight phase shift between clock and data signal due to various reasons. Hence method shown in Fig 4.4 is ideal way to generate manchester encoded signal. It was experimentally observed that we get short impulses due to phase mismatch in manchester encoded data output of circuit as shown in Fig 4.6. Hence we need more sophisticated way of generating manchester encoded data. Once such method is shown in Fig 4.5.

Circuit shown in Fig 4.5 is based on two D flip flop and three 2-1 multiplexers. In this circuit, incoming data is latched with the help of D flip flop and this data is inverted. This synchronized data and inverted data is selected using 2-1 multiplexer alternatively with timing provided by clock. Output signal out of multiplexer is manchester encoded data stream. In this circuit IC7474 D flip flop and IC74157 quad multiplexer is used. Output of first flip flop is latched or synchronized data while output of second flip flop is inverted data. This two signals are routed through 3rd multiplexer of 74157. Using clock as a select signal for this multiplexer, we select between data and inverted data alternatively. Output of this multiplexer is nothing but manchester encoded data.

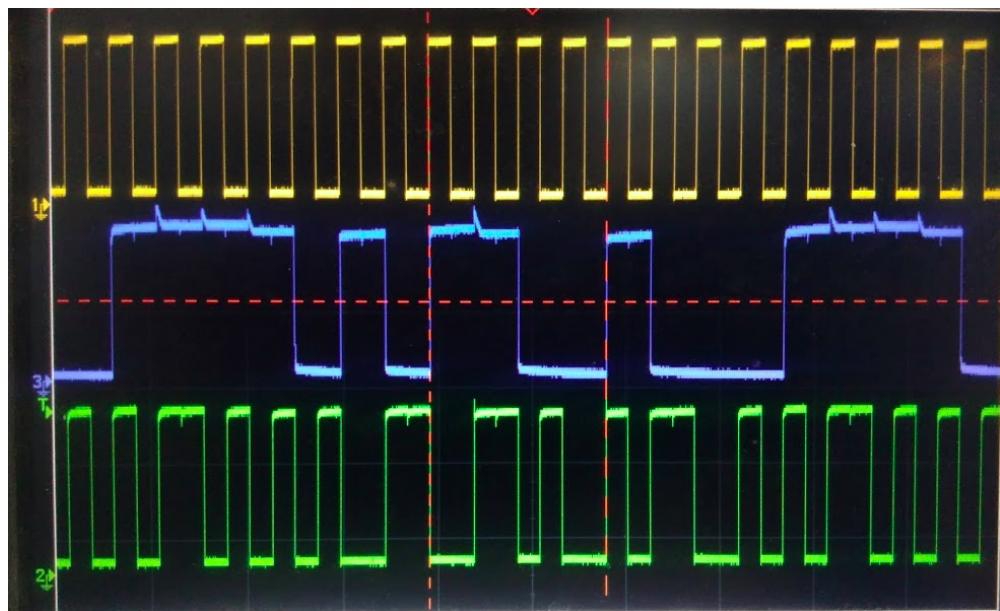
This circuit was tested and results are as shown in Fig 4.7. Where signal in green is clock, signal in blue is PRBS data and signal in pink is manchester encoded data. This circuit was tested at 1MHz of frequency.



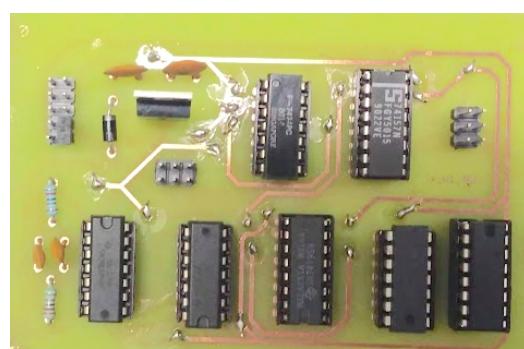
**Fig. 4.5.** Manchester encoder



**Fig. 4.6.** Manchester encoded waveforms using 4.4



**Fig. 4.7.** Manchester encoded using 4.5



**Fig. 4.8.** Transmitter PCB

## 5 Clock and Data Recovery(CDR)

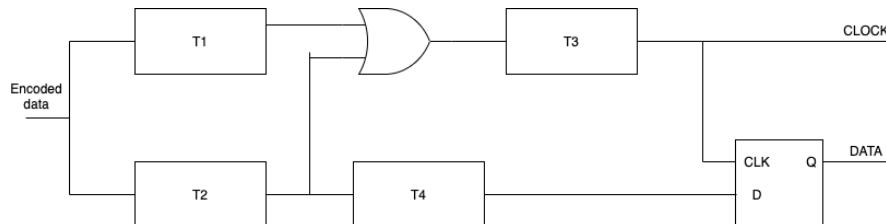
This block extracts clock and data from received manchester encoded data. There are various methods for clock and data recovery. First is using digital phase locked loop and other using monostable multivibrators.

### 5.1 CDR using one shot

Fig 5.1 shows the logic of clock and data recovery from monostable multivibrators. 4 one shots 74LS121 and 1 D flip flop 74LS74 are used in this circuit.

two monoshots T1 and T2 are used as transition detector. T1 is used as a positive transition detector. T2 is used as negative transition detector. These transitions contain timing information. As we are using manchester encoding we are sure that during each clock cycle we have at least one transition in the middle. But if we have consecutive 1's or 0's in the data stream, we get two transitions in that particular clock period. Hence this extra transition has to be removed in order to extract clock. Output of OR gate is total transitions(including both detected by T1 and T2), now time constant of T3 is chosen such that this extra transition pulse is eliminated. T3 is positive edge detector. We require T3 value as given in equation.

$$0.5T < T3 < T \quad (1)$$



**Fig. 5.1.** Clock and data recovery

We can see that this circuit is frequency dependent. Output of one shot T3 is required clock signal. Once we get clock we can extract data using another negative edge detector monoshot T4 and D flip flop as shown in the circuit. Time constants T1,T2,T4 should satisfy following equations.

$$T1 < T \quad (2)$$

$$T2 < T \quad (3)$$

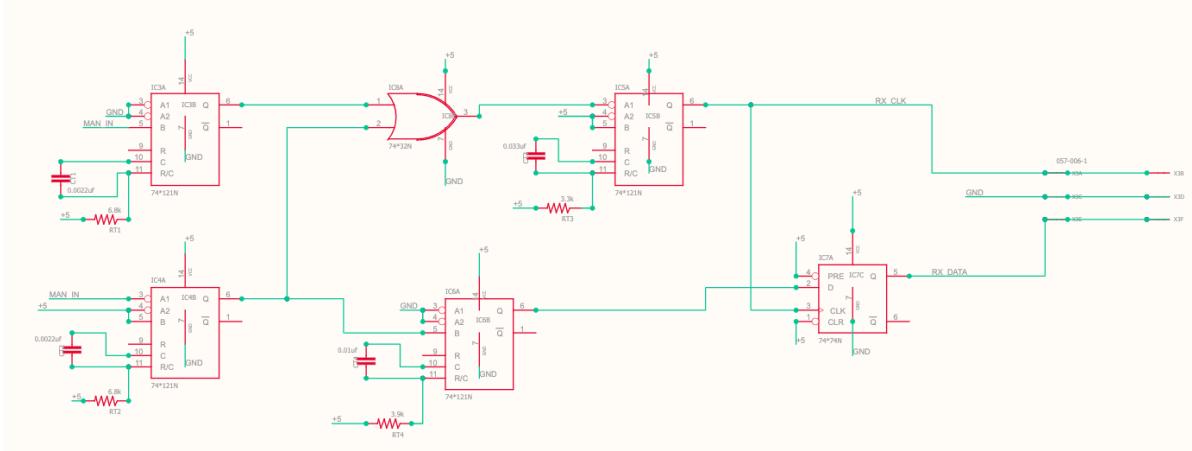
$$T1 < T4 < 0.375T \quad (4)$$

It is obvious that this circuit is frequency dependent. Hence values of these all parameters has to be chosen wisely for various working frequencies. However once designed for particular frequency, this circuit can operate correctly for 33 percent frequency variation.

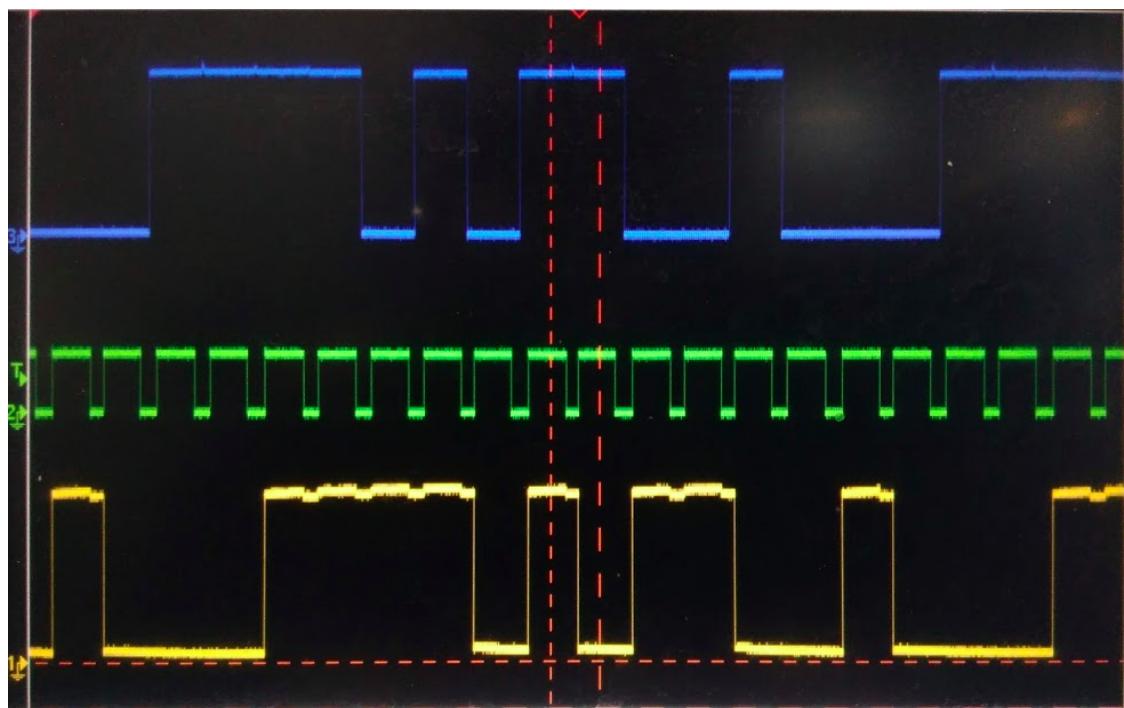
The time constant of each one shot is determined by following equation, where R and C is value of Resistance and capacitor.

$$T = 0.69RC \quad (5)$$

Fig 5.2 shows the eagle schematic for this CDR circuit.



**Fig. 5.2.** Clock and data recovery



**Fig. 5.3.** CDR waveforms

Fig 5.3 shows the testing of this circuit. Signal in blue is transmitted PRBS data. Signal in green is extracted clock from received data. Signal in yellow is extracted data. This testing was done at 10KHz frequency.

## 6 BER checker

Once we decode the manchester encoded data, we get clock and data sepearte. Now we need to check the validity of this data. This is done by BER checker circuit as shown in Fig 6.1. We use IC74195 4bit shift register as a buffer to store incoming data bits.

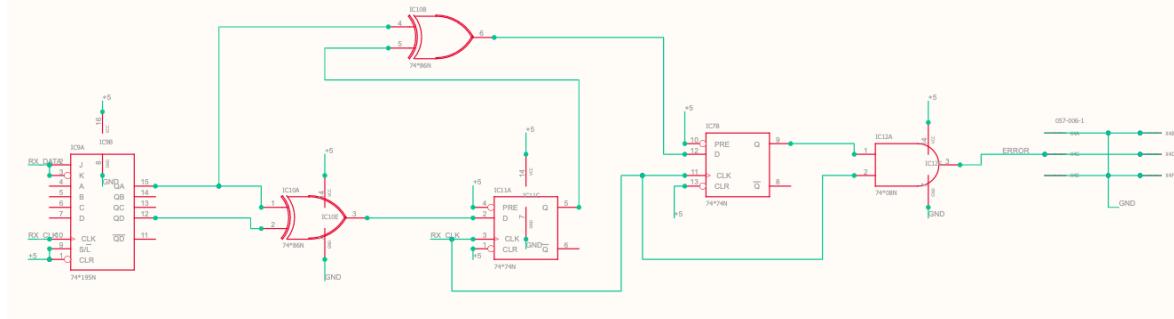


Fig. 6.1. Error pulse generation

These data bits are represented as Qa, Qb, Qc and Qd. As we know the method we generate PRBS data at transmitter we can use same method to generate new data bit locally. Consider initially all bits in Qa, Qb, Qc and Qd are valid. Then next bit that we expect is Qa xor Qd. Hence we calculate Qa xor Qd using Exor gate IC10A and store or latch this using D flip flop so that we can compare this data with one we actually receive in next clock cycle. Hence we compare locally generated bit and received bit using ExOr gate IC10B. This compared data is latched and gated with AND gate IC12A. When we receive the correct data output of comparator ExOr will be logic 0, hence we get output as logic 0, when there is error, output of comparator is logic 1. Then at output we get pulse with pulse width equal to width of clock signal. Hence at the output of this circuit we can get a pulse corresponding to each error. Now we count these pulses in specific time period using counters and then calculate Bit Error Rate(BER). Fig 6.2 shows error pulses in green. While checking this circuit we intentionally make errors by placing obstacle between transmitter and receiver. We test Manchester decoder and BER testing circuit on breadboard as shown in Fig 6.3

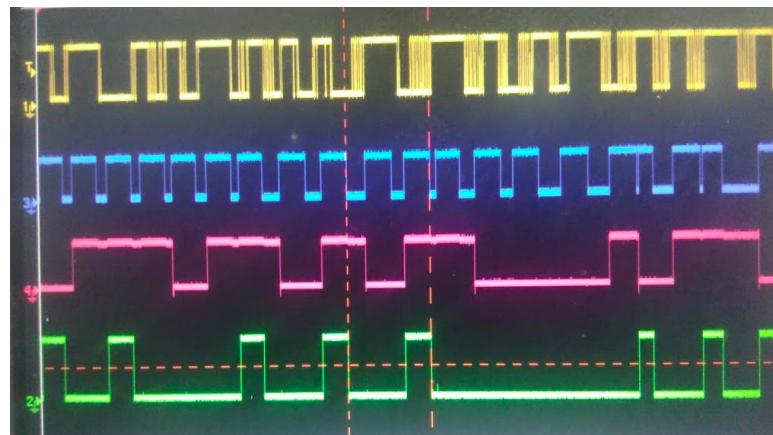


Fig. 6.2. Manchester encoded using ??

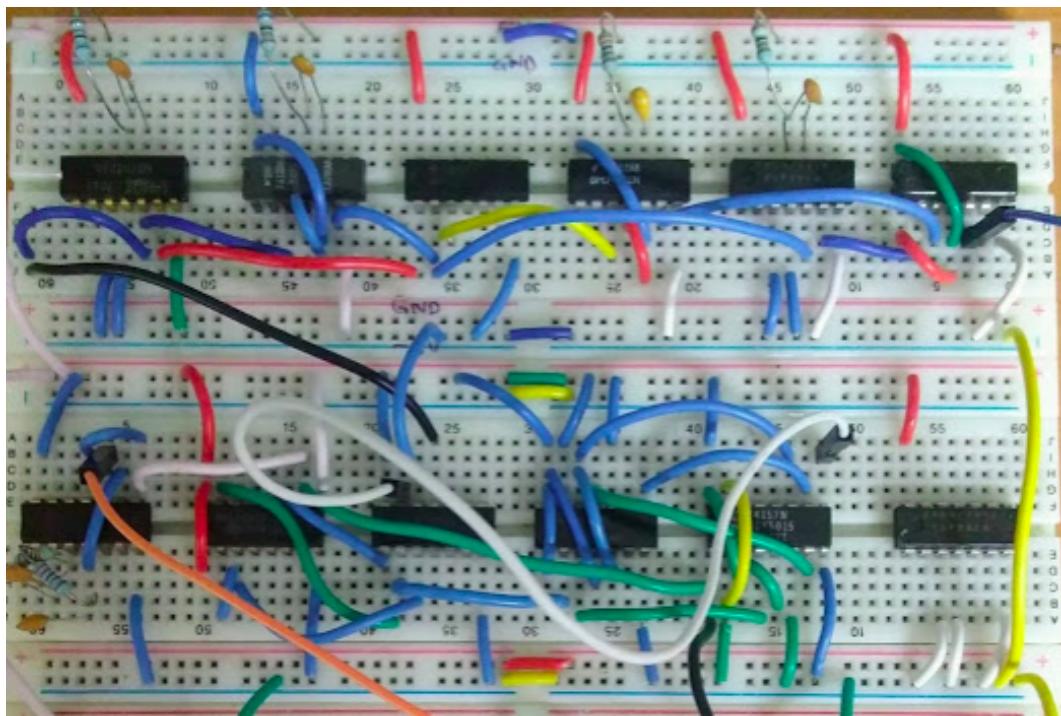


Fig. 6.3. Testing on bread board