TCOM370

Homework Set 3 Solution

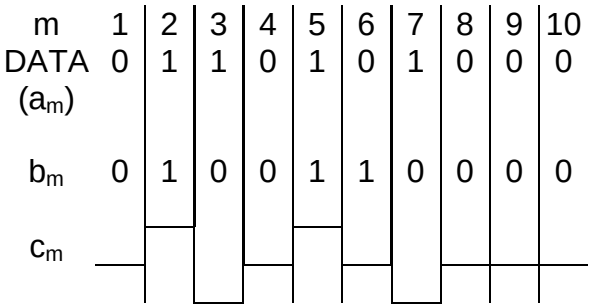
Problem 1

The number of bits per character prior to transmission is 11 bits, of which 7 are information bits, and the other 4 bits are transmission over head. Thus only $\frac{7}{11}$ of the transmitted bits are useful information. Since there are 2 bits per signaling element (by using 4-phase PSK, for example), the bit rate is twice the baud. The number of information bits in each case is thus as follows:

Baud (Signaling) rate	Bit rate (bps)	Information rate (bps)
300	$300 \times 2 = 600$	$600 \times \frac{7}{11} = 382$
600	$600 \times 2 = 1200$	$1200 \times \frac{7}{11} = 764$
1200	$1200 \times 2 = 2400$	$2400 \times \frac{7}{11} = 1528$
4800	$4800 \times 2 = 9600$	$9600 \times \frac{7}{11} = 6112$

Problem 2

Before we dive in to the problem, consider a concrete example to make sure you understand what is happening here. Suppose the sequence 01101010000 is transmitted (a_m) and b_0 is "0".



Observe that $c_m=0$ when $a_m=0$, and $c_m=\pm 1$ (alternating between the two) when $a_m=1$. Since the receiver receives c_m, it must decide what data was sent from c_m only. The decision rule given in the problem is "d_m='0' if received pulse amplitude is close to 0; otherwise d_m='1'". Clearly, this decision rule would map ±1 to "1" and 0 to "0."

(a) $d_m=a_m$

Note that because $b_m = (a_m + b_{m-1}) \pmod 2$, the b_m changes logic state every time a_m="1". Thus by looking at the arithmetic differences $c_m = b_m-b_{m-1}$ we can detect where the a_m="1" (the c_m will be +1 or -1). Also note that the change in logic state of the b_m can be from 1→0 or 0→1, these changes alternate whenever they occur. This leads to the answer in part (b).

(b) AMI (alternate mark inversion)

(c) Recall that pseudoternary line code is the opposite of AMI, i.e. encode 0's with alternating polarity pulses and send 0's with a zero level. Therefore, am should be inverted prior to encoding:

$$b_m = (\overline{a_m} + b_{m-1}) \bmod 2$$
$$c_m = b_m - b_{m-1}$$

Note $\overline{a_m}$ equals the logical inverse of a_m, where "0" becomes "1," and "1" becomes "0". To decode the sequence use the following rule: d_m= "1" if the received pulse is close to zero and d_m= "0" otherwise.

Problem 3

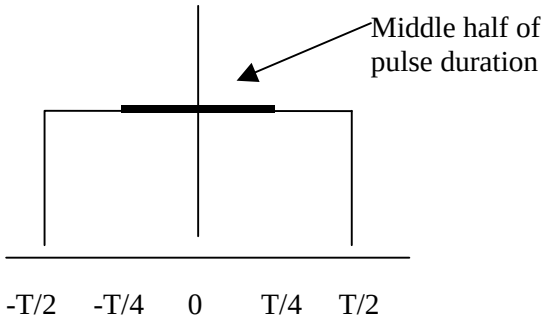
Recall that both Manchester and differential Manchester codes must have transitions in the middle of the bit period. Using this information we can find out which T/2 edges correspond to the middle of the bit period. Clearly there are two possibilities only for the T-length bit intervals on this figure. The very first up-transition at the first tick cannot be a bit-interval edge because it would lead down the time axis to a situation with a bit-interval having no transition in the middle

(a) 1 1 1 1 0 0 1 1 0

(b) In differential Manchester coding, the type of the transition for the bit being transmitted stays the same as that of the preceding one if the current bit is 0, and switches if the current bit is 1.
So the two possible bit streams are 0 0 0 0 1 0 1 0 1 and 1 0 0 0 1 0 1 0 1.
Note that it is not possible to determine what the first bit is since we do not have knowledge about the previous bit.

Problem 4

A data bit is received correctly if its waveform is sampled in the middle half of its duration as shown below:



There are several ways of thinking about this problem. We'll think about it 2 different ways.

1) It is clear from the picture above that drift of T/4 seconds is the maximum the transmitter/receiver can tolerate before incorrect samples begin to be taken. Since both the transmitter and receiver clocks are drifting, at the worst case, the two clocks drift in opposite directions, and the effective drift of 1 ms/minute is achieved (twice the value for each clock).

   Bit period = 1/10 Mbps = 100 ns = 0.1 μs

   Effective drift = 1 ms/minute = 0.0167 ms/s = $0.0167\frac{ms}{s}\times\frac{0.1ms}{bit}=1.67\times10^{-12}\frac{s}{bit}$

Since the system can tolerate up to ¼ of a bit (T/4 = 0.025 μs), we can calculate the number of bits before synchronization is needed: $\frac{0.025ms}{1.67\times10^{-12}\ s/bit}=15000bits$

2) From the picture above, it is clear that the system can tolerate up to T/4 second of drift between the transmit clock ticks and receive clock ticks. However, we must remember that both the transmitter and the receiver are subject to drift. In order to account for the worst case scenario, we must assume that the clocks at the transmitter and the receiver are drifting in the opposite direction (i.e. one is getting faster while the other is getting slower than the target clock rate). After only T/8 seconds of drift, there is effectively T/4 seconds of offset, and incorrect samples begin to be taken.

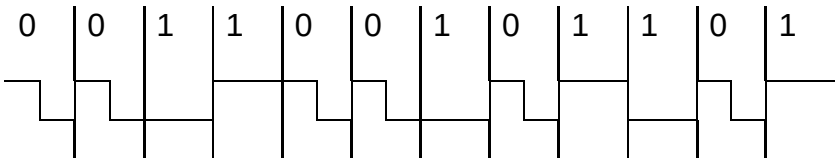
   Clock rate = 10 MHz $\Rightarrow$ T = 1/10 MHz = 100 ns $\Rightarrow$ T/8 = 12.5 ns
   Drift = 0.5 ms/minute = 8.333 μs/s

The time to reach T/8 of drift is $\frac{12.5\text{ns}}{8.333\ \ ms/\text{s}} = 1.5\text{ms}$. Since we are transmitting at 10 Mbps, we can transmit up to 15000 bits before we lose synchronization and have to re-synchronize.

Note that actually the transmit data rate is not exactly 10 Mbps, because if the transmit clock is faster or slower it will change the transmit rate. However, since the offset is no more than 0.5 ms per min for the transmit clock, the max and min rates are 10x60 Mb every (1 min - 0.5 ms) or every (1 min + 0.5 ms), which are very close to 10 Mb per sec which is the value used to get the answer above. The synchronization error will kick in way before the slight difference in transmit rate from nominal has a chance to have any efffect on the answer.

Problem 5

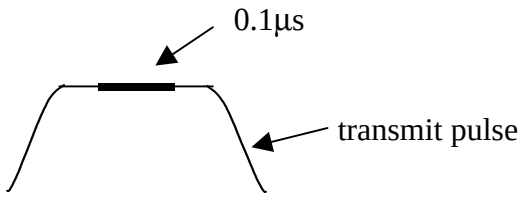
(a)



(b) The longest duration of a constant level is 3T/2. The shortest duration of a constant level is T/2.

(c) Bandwidth – 2/T (same as Manchester)
DC value – If the levels are chosen appropriately (opposite polarity and same amplitude), the DC component is zero in both Manchester and CMI code.
Timing recovery – Clock recovery is possible since all "0" bits have transitions in the middle (ala Manchester coding) of the bit period, and "1" bits have transitions at the ends of the bit interval There is a transition at least every 1.5 bit periods.
Polarity inversion – CMI code can recover but Manchester cannot (differential Manchester can, however)

Problem 6 (Problem 2 from 1998 Exam 2)
A data bit is received correctly if its waveform is sampled within the middle 0.1μs as shown below:



If transmit and receive clock ticks offset $\frac{0.1}{2} ms$ by or more, errors start occurring.

Maximum rate of relative drift = 2×0.6 ms/minute = $\frac{1.2}{60}$ ms/s = $0.02 ms/s$

$\therefore$ In τ seconds, offset is 0.02τ ms.

Set $0.02\tau = \frac{0.1}{2} \times 10^{-3} ms$;

$\therefore\ t = 2.5 \times 10^{-3} \sec s = 2.5 ms$