Data Communication for Rules

PAGE
DATE

(1) Sine wave :-

$$S(t) = A \sin(2\pi f t + \theta)$$

(A) → peack Amplitude (Maximum strength of signal) → (V)
* In frequency-domain plot, the vertical axis measures the (peack Amplitude)

(f) → Rate of change of signal → (Hz)
* period :- time for one repetition (T) = (1/f) → (Decreases)

(θ) phase :- Relative position in time.

(λ) wavelength :- Distance between two points of corresponding phase in two consecutive cycles.

$\lambda = CT \rightarrow \lambda = C\,1/f \rightarrow \lambda f = c \;/ c = 3 * 10^8$ m/s

Bandwidth rule :- → $f_{max} - f_{min}$

Data rate = no. of. bits / total bit duretion → bit duretion = total bit duretion / no. of. bit

bit rate = 1 / bit duretion → no. of. bits = $\log_2 L$ (L → no. of levels)

Attenuation →

$$N_{dB} = 10 \log_{10} \left(\frac{P_{out}}{P_{in}} \right)$$

P_{out} → output power
P_{in} → input power

Channel Capacit

Nyquist Bit rate

$C = 2B \log_2 M$ (M → signal levels)

→ Channe
→ bit rate
→ data rate

Shannon Capacity

$C = B \log_2 (1 + SNR)$

SNR → Signal to noise ratio

PAGE

DATE

$$SNR = \frac{\text{Signal power}}{\text{noise power}} \rightarrow SNR_{dB} = 10 \log_{10} SNR$$

$$SNR = 10^{SNR_{dB}/10}$$

$$\text{Propagation time} = \frac{\text{Distance}}{\text{Propagation speed}}$$

The Relationship between propagation time and Propagation speed

$$\text{Transmission time} = \frac{\text{Message length}}{\text{bit rate}}$$

$$\text{Troughput} = \frac{\text{Avg no. of form} * \text{form size}}{\text{Time (s)}}$$

* Key paramaters of anttenna Gain:-

$$G = \frac{4\pi A_e}{\lambda^2} \rightarrow \lambda = \frac{f}{C} \rightarrow G = \frac{4\pi f^2 A_e}{C^2}$$

A_e : effective area (m^2)
λ : Carrier wavelength (m)
f : Carrier of frequency (Hz)
C : speed of light ($c = 3 \times 10^8$ m/s)
G : antenna Gain (ratio)

$$G_{dB} = 10 \log_{10} G \rightarrow G = 10^{G_{dB}/10}$$

* Line of sight (LOS)

$$d = 3.57\sqrt{Kh}$$

(Constant = 4/3)

distance

height of antenna (2)

$$d = 3.57\sqrt{Kh_1} + \sqrt{Kh_2}$$

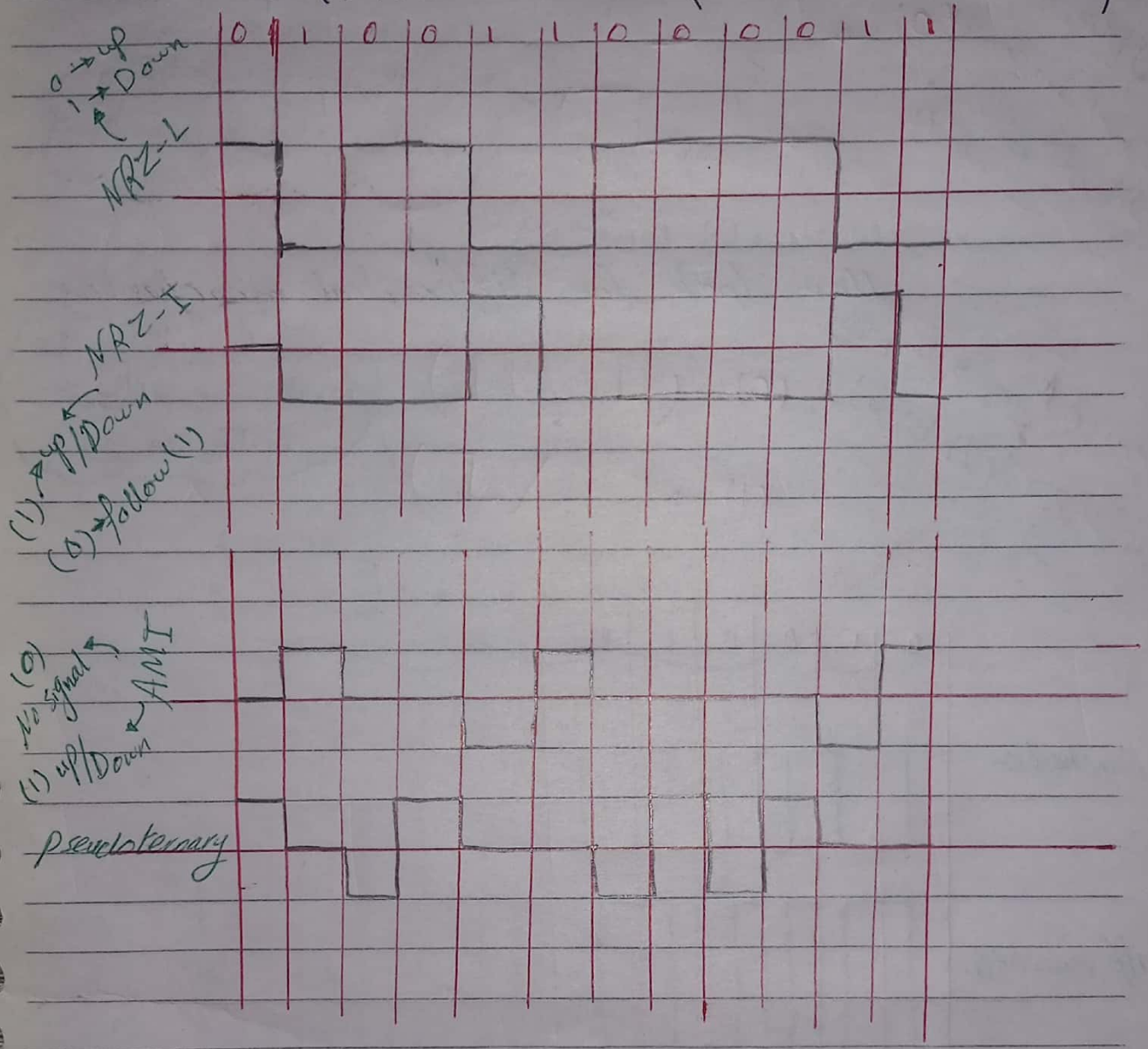
height of antenna (1)

PAGE

DATE

Convert from digital data to digital signals

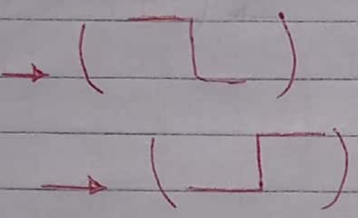
1- NRZ-L \ NRZ-I → (0 1 00 11 0000 11)



Look here for manchester

بص يا نجم

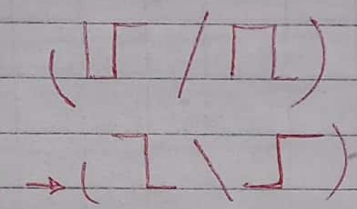
هنا يا زميلي ال(0) لازم يبدأ من فوق لتحت خلي بالك

(0) →

(1) →

هنا خويا لازم يكون ال(1) بادئ من تحت لفوق لازم

Now look for differential manchester

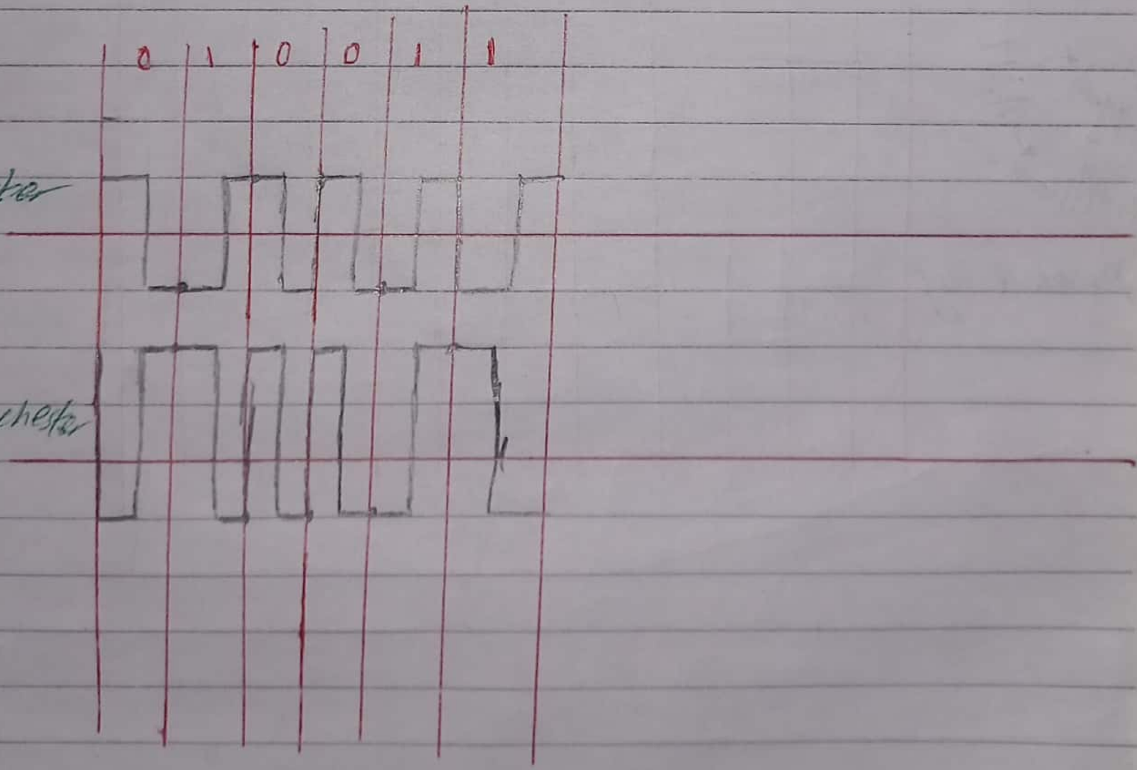
هنا بقا يجي ال(0) لازم يكون (2bits) أياً كان شكله

(0)

(1) →

هنا يعم بقا بيبدأ من تحت لفوق من فوق لتحت مش فارقة

manchester

diff manchester



PAGE

DATE

*B8ZS → (هنا يعلم ما يتشخلش الا لو في (8 Zero

في (8 Zeros) وقبلهم (+) يعني up هنطبع (+ - 0 - + 000) ←

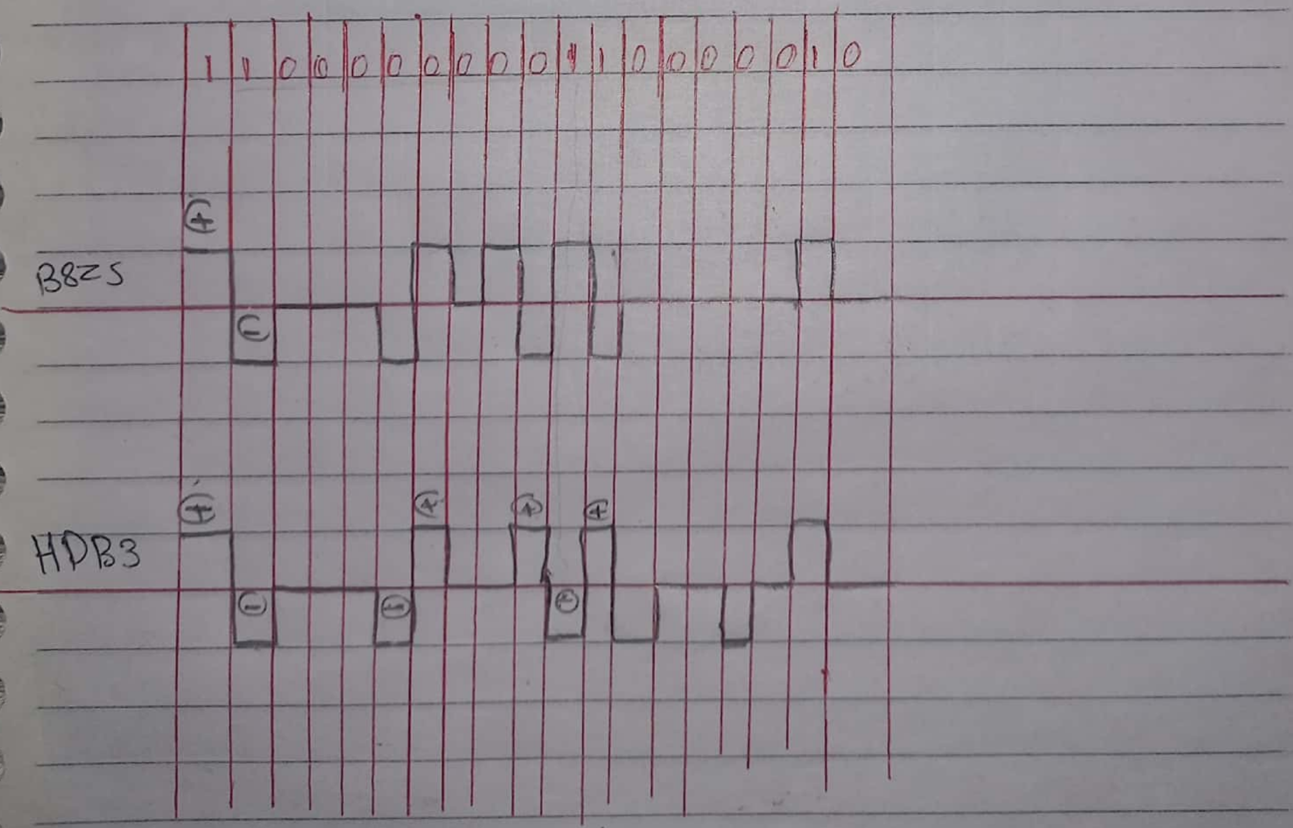
في (8 Zeros) وقبلهم (-) يعني Down هنطبع (- + 0 + - 000) ←

← مفيش (8 zeros) هنطبعه الـ AMI واي (0) بـ No signal

*HDB3 → هنا بالكبير مش هتشتغل إلا لو في (4 Zeros)

← لو اللي قبل الـ (4 Zeros) (-) عددها (odd) هنطبعه (- 000)
← " " " " " عددها (even) هنطبع (+ 00 +)

← لو اللي قبل الـ (4 Zeros) (+) عددها (odd) هنطبعه (+000)
← " " " " " عددها (even) هنطبع (- 00 -)



PAGE
DATE

$r = N/S$

- N → no. of bits per seconds
- S → no. of elements per second
- r → Baud rate

Buad rate is the number of signal elements per second.

In the analog transmission of digital data, the baudrate is less than or equal to the bit rate

---

Sampling rate = 2B → (Sampling/sec)

القوانين دي بتستخدمها لما يديلك مسألة على الـ (PCM) ويطلب منك المطاليب دي

Data rate = no. of bit * sampling rate

S_{NR} = 6.02 + (no. of bits) * 1.76

no. of Cells = $\frac{\text{bit rate for all system}}{\text{bit rate for one cell}}$

- bit rate for all system → Given
- bit rate for one cell → هو نفسه الـ Data rate

Baud rate = Bw = $(f_1 - f_2)$

- Bw → Bandwidth
- $(f_1 - f_2)$ → difference between two Carries

bitrate = baud rate * no. of bit per second

* القوانين دي بنستخدمها لما يدينا الـ (Bw) وكلام بيدينا الـ $(f_1 - f_2)$
وغالباً بتبقى لـ (FSK signal)

خد بالك لو بيدينا الـ Bw ← (full duplex)

وأي full duplex ← بنقسمها على (2)

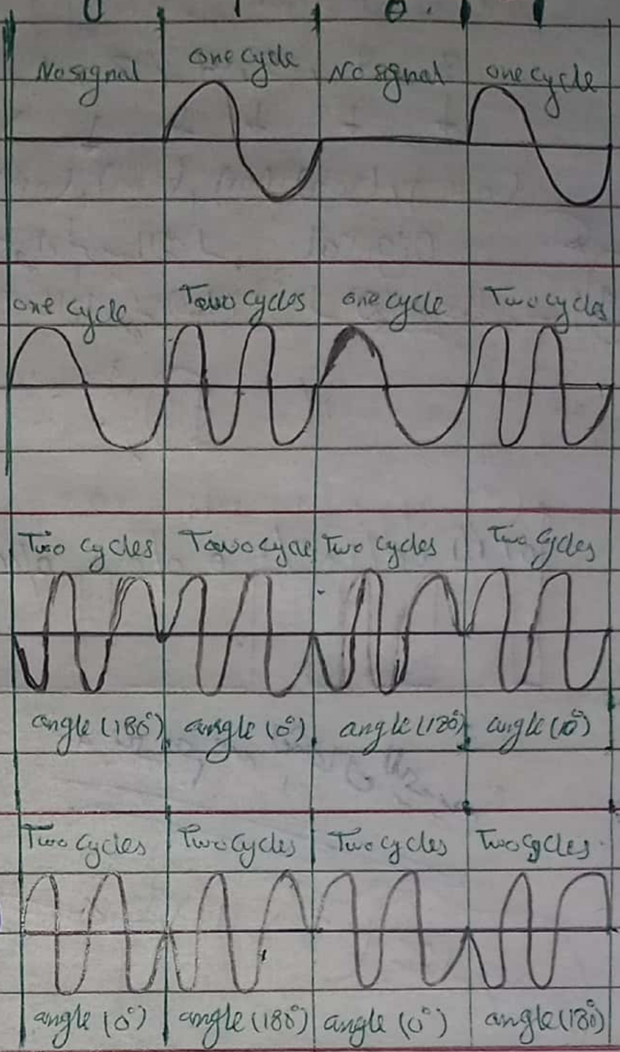
Convert from digital data to analog signal :-

1- ASK → (0) No signal
→ (1) one cycle
~~start~~ with angle (90°)

2- FSK → (0) one cycle
→ (1) Two cycles
~~start~~ with angle (90°)

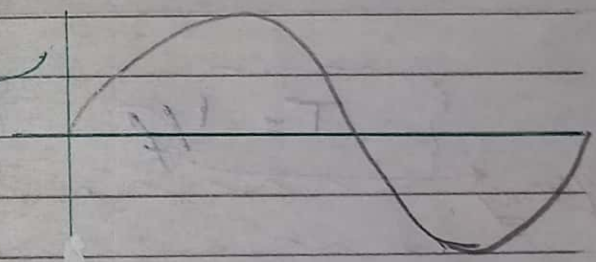
3- PSK → (0) → Two cycles
(1) → Two cycles
(0) → start with angle (180°)
(1) → start with angle (0°)

4- BPSK → (0) → angle (0°)
→ (1) → angle (180°)
Both of them have two cycles.



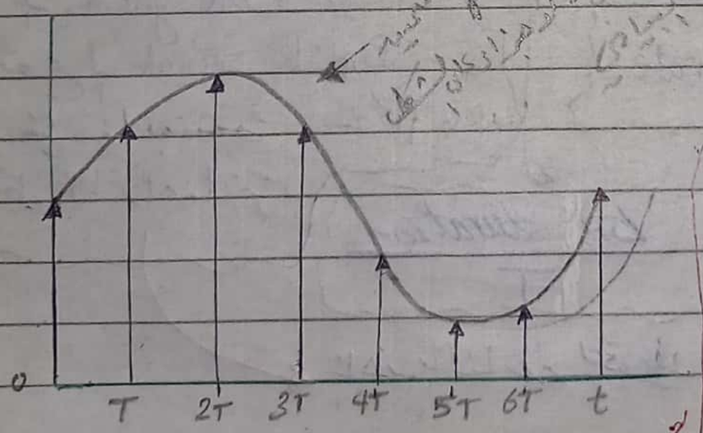
* Steps of PCM

Analog Signal



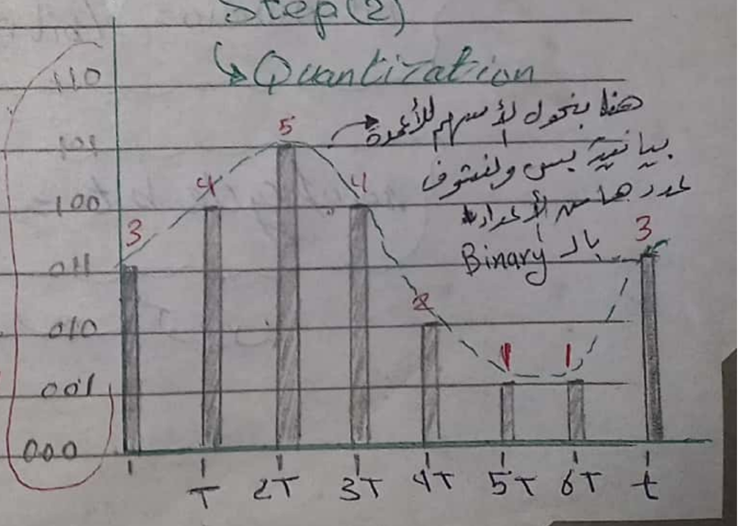
Step (1)
↳ sampling

هنا احنا بنقسم الـ signal لأجزاء صغيرة



Step (2)
↳ Quantization

هنا بنحول الأسهم للأعمدة بيانية بس وبنشوف عدد هاصم الأعداد بالـ Binary



الأرقام هنا بالـ Binary يعني
(011 = 3)

Step(3) Coding

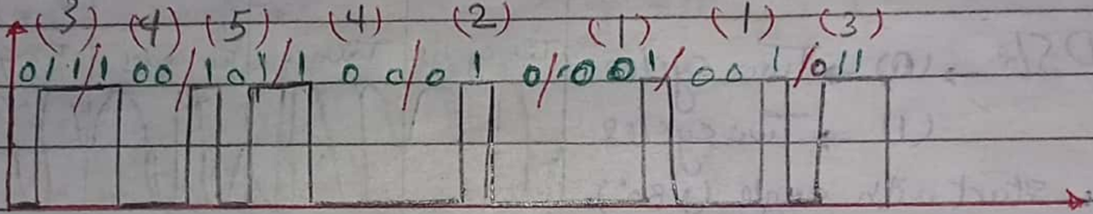
هنا بناخد الرقم بال Binary من الشمال لليمين الترتيب بقى

3 , 4 , 5 , 4 , 2 , 1 , 1 , 3

(011) , (100) , (101) , (100) , (010) , (001) , (001) , (011)

بعد كدا بنبدأ نحول الرسالة لـ Digital

وناخد بالنا ركزها إن الـ (0) بيتمثل بـ (flat) والـ (1) بـ (pulse)



بنرسم من الشمال لليمين

كل (3 bits) تمثل رقم

$T = 1/f$ (frequency)

bit duration = 1/bit rate

$$\text{no. of cycle bit} = \frac{\text{bit duration}}{T}$$

بنستخدم القوانين دي في المسألة لما بيدينا bit system و بيدينا نوع الـ BPSK signal فإحنا بنشوف فيه كام cycle لكل bit وإحنا بنرسم

ناخد بالنا من التحويل ده كويس

* "parity check

* Odd → بنعد عدد الواحدات في الـ Message لو كانوا (odd) نضيف (1) للـ Message من جهة اليمين

* Even → بنعد عدد الواحدات في الـ Message لو كانوا (even) نضيف (0) للـ Message من جهة اليمين

11001 0011

odd (1)

11001 00111

00111011111

even (1)

001110111110

EX: The bit stream 10011101 is Transmitted using the CRC method described in lecture using generator Polynomial x^3+1

1) show the bit string transmitted

2) show that, if the third bit from left is inverted during the Transmission, the receiver Can detect it.

Sol: $M = 10011101$, $P = x^3+1 = 1001$

Scanned with CamScanner

1) Transmetter:

① Add no. of zero to message: — اول حاجة هنضيف عدد من الاصفار ، عدد هم بيكون على الاكبر degree للـ X ، هنا عندي x^3 ← على degree

$M' = 10011101000$

② Divided M` by P — هنقسم الـ M` على P ،

```
           10001100
1001 ) 10011101000
       1001↓
       x0001
        0000↓
        x0011
         0000↓
         x0110
          0000↓
          x1101
           1001↓
           x1000
            1001↓
            x0010
             0000↓
             x0100
              0000
reminder → x100
```

اول حاجة هنبص على على اول bit على شمال ونحطها زي ماهي فوق ، بعديها ناخد الـ bit اللي حطيناها فوق ونضربها بالـ P ناتج الضرب هنحطه تحت الـ M` بعدين هنعمل XOR بين اول bit على الشمال مش هنعملها XOR ونحط مكانها في الناتج (X) بعد ماعملت XOR هننزل bit من الـ M` اللي تحت مع الناتج وهنفضل كدا لحد ماخلص الـ bit stream كله

③ merge reminder to message:

$T = 10011101100$

بعد ماخلصت الـ bit stream هنضيف ناتج القسمة اللي هو الـ reminder على الـ M اللي هي الحاجة الأصلية

2) Receiver:

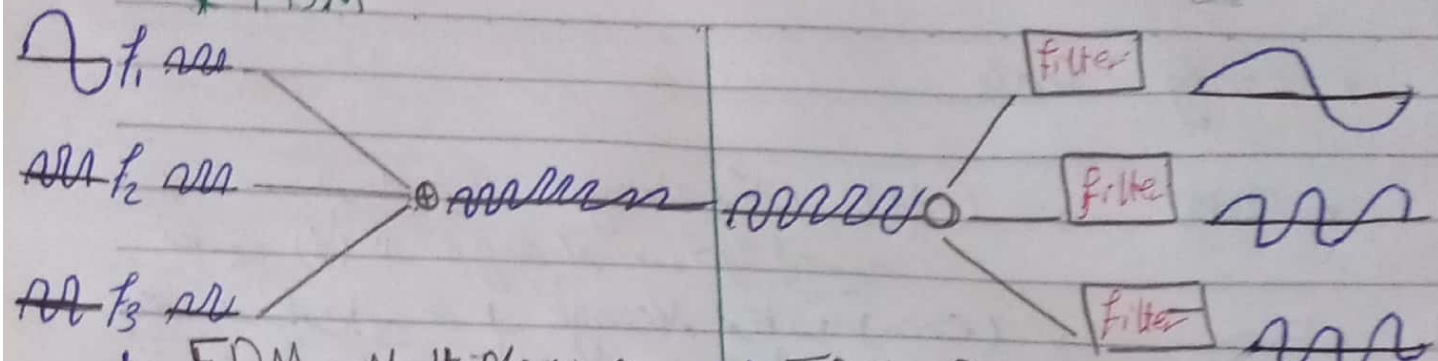
```
          10101000
1001 ) 10111101100
       1001↓
       x0101
        0000↓
        x1011
         1001↓
         x0100
          0000↓
          x1001
           1001↓
           x0000
            0000↓
            x0010
             0000↓
             x0100
              0000
              x100 → reminde ≠ error
```

هاخد الـ T اللي طلعتها فوق واقسمها على الـ P بس خلي بالك هنا في السؤال بيقولك ان ثالث bit ناحية الشمال حصلها invert وبالتالي هغير قيمتها لو (1) هخليها (0) ولو (0) هخليها (1) وطريقة القسمة زي اللي فوق

مفيش reminder يبقى مفيهوش error ودا طبيعي لان احنا غيرنا bit

CamScanner

* FDM



(FDM - Multiplexing) (FDM - DEmultiplexing)

no. of guard band = no. of signals - 1

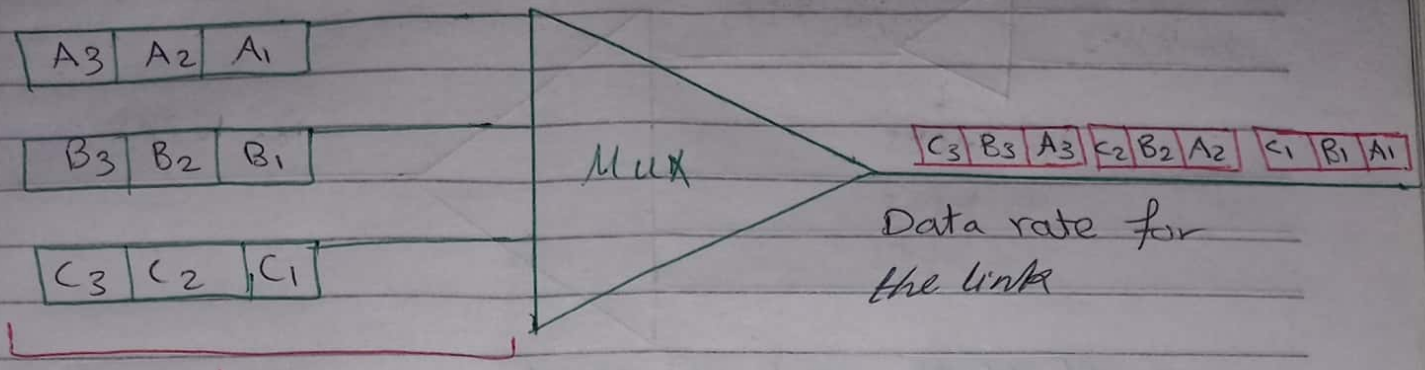
Min Bandwidth for Multiplexing =

no. of signals * Bandwidth of one signals + no. of guard band * Bandwidth of guard band

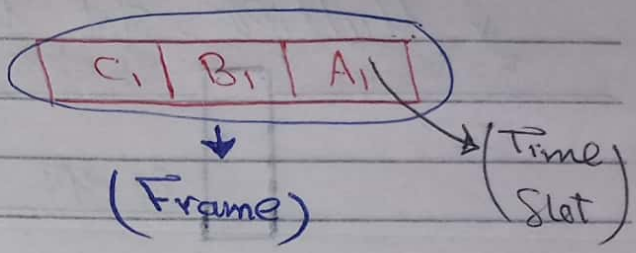
لي مسألة بتيجي فيها FDM multiplexing وبيقا مديلك
- no. of signals
- Bandwidth of one signal
- Band widt of guard band

بس يا معلم ركز بقا وانت بتحل .

* TDM



Channel rate / input rate



Frame Size = no. of Channels * Time Slot

Fram duration = 1/frame rate

Fram rate = Channel rate / Time Slot

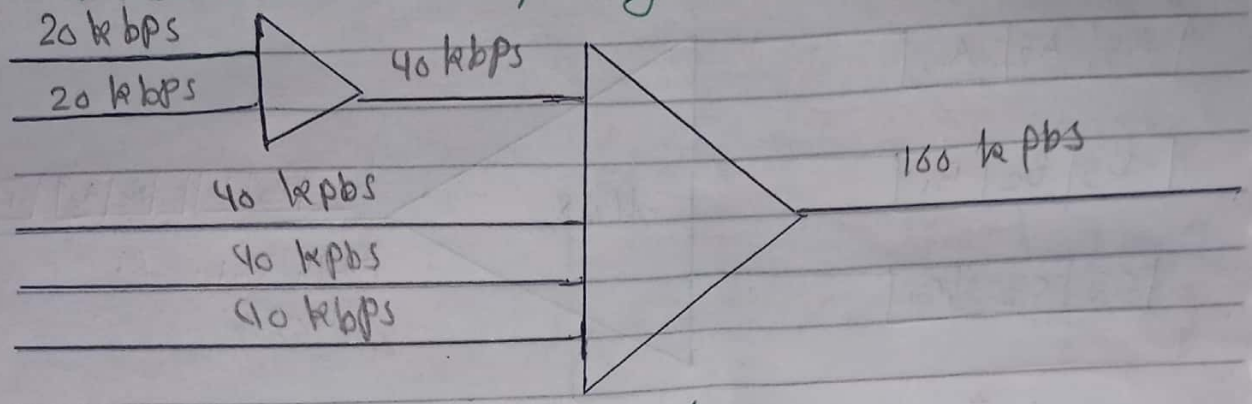
output bit rate = Channel rate

Output Data rate = Channel rate * no. of Channels

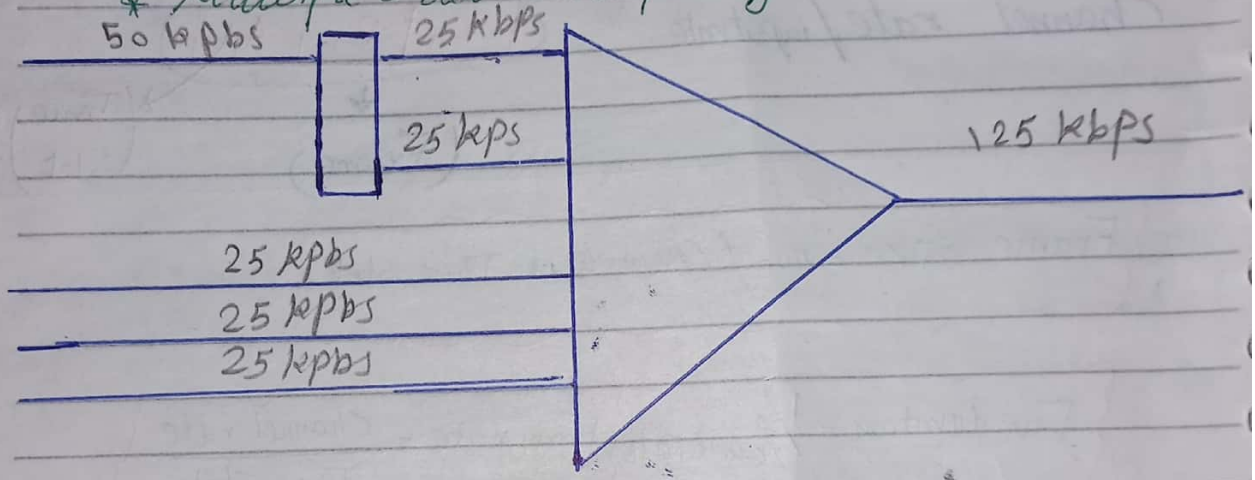
output bit duration = 1 / output Data rate

PAGE
DATE

* Multilevel multiplexing :-



* Multiple - slot Multiplexing :-



* Pluse stuffing