ADITYA ENGINEERING COLLEGE(A)

(Approved by AICTE, Accredited by NAAC with 'A' Grade, Recognized by UGC under the sections 2(f) and 12(b) of UGC Act 1956)

LABORATORY MANUAL

DIGITAL COMMUNICATIONS LAB

Regulations : AR 20

Course Code :

Semester : V

Department of

Electronics and Communication Engineering

Prepared by
Dr.U.Rajyalaxmi, Prof

Aditya Nagar, ADB Road, Surampalem, East Godavari District, Andhra Pradesh

VISION & MISSION OF THE COLLEGE

VISION

To induce higher planes of learning by imparting technical education with International standards, Applied research, Creative ability and Value based instruction to emerge as a premier institute.

MISSION

Achieving academic excellence by providing globally acceptable technical education by forecasting technology through

- Innovative research & development
- Industry institute interaction
- Empowered manpower

VISION & MISSION OF THE DEPARTMENT

VISION

To empower the Electronics and Communication Engineering students with technological capability, professional commitment and social responsibility.

MISSION

M1: Providing quality education through dedication to duty, best of breed laboratory facilities, collaborative ventures with the industries and effective teaching-learning process.

M2: Pursuing research and promoting new technologies in order to serve the needs of the society, industry, government and scientific community.

M3: Equipping the students with strong foundations to enable them for continuing education.

****

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of the Program will

PEO 1	Adapt the learning culture needed for a successful professional career by encouraging them to acquire higher qualifications, take up research and keep abreast of latest technological developments.
PEO 2	Design, build, and test electronic and modern communication Systems for the given specifications by considering technical, environmental and social contexts.
PEO 3	Communicate effectively, demonstrate leadership qualities and develop knowledge of societal impacts of communication technologies with professional ethics.

PROGRAM OUTCOMES (POs)

After successful completion of the program, the graduates will be able to

PO 1	Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
PO 2	Identify, formulate, research literature and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO 4	Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO 5	Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.
PO 6	Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
PO 7	Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of, and need for sustainable development.
PO 8	Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
PO 9	Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO 10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	Demonstrate knowledge and understanding of engineering management principles and apply these to one's own work, as a member and leader in a team and to manage projects in multidisciplinary environments.
PO 12	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technologicalchange.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After successful completion of the program, the graduates will be able to

PSO 1	Apply the knowledge of Analog & digital system design concepts in the fieldof Electronics and Communication Engineering practice.
PSO 2	Design and conduct experiments, as well as analyze and interpret the data toarrive at substantial conclusions using CAD tools: Cadence,P-Spice, Xilinx ISE,MATLAB, Mentor graphics
PSO 3	Design an electronic system or process to meet the real life and societal needs.

ADITYA ENGINEERING COLLEGE(A)

Approved by AICTE, Affiliated to JNTUK & Accredited by NAAC with 'A' Grade
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Aditya Nagar, ADB Road, Surampalem – 533 437

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ANALOG COMMUNICATIONS LAB

GENERAL INSTRUCTIONS

Do's

- Wear your ID card while entering the lab.
- Keep your laboratory clean.
- Maintain silence inside the laboratory.
- Report immediately to the lab-incharge if there is any problem identified in hardware/software.
- After completing the experiment, Equipment/accessories should be handed over to the Lab Assistant.
- Experiments should be conducted under the supervision/permission of faculty/staff.
- While leaving the lab, keep the chairs in the proper position.

Don'ts

- Don't use cell phone inside the laboratory.
- Don't throw any waste on the floor.
- Don't stick or write anything on the walls or furniture'

HEAD OF THE DEPARTMENT

ADITYA ENGINEERING COLLEGE(A)

(Approved by AICTE Accredited by NAAC with 'A' Grade
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Aditya Nagar, ADB Road, Surampalem – 533 437)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ANALOG COMMUNICATIONS LAB

SAFETY MEASURES

Potential Problems:

- Electric Shock
- Explosion of compoentnts

Safety precautions to minimize these hazards:

- Keep the work area neat and clean.
- Use rubber floor mats (if available) to insulate yourself from ground, when working in the Lab.
- Be sure about the locations of fire extinguishers and first aid kits in lab.
- Use only isolated power sources.
- Check any connections for shorting two different voltage levels.
- Double-check your wiring and circuit connections. It is a good idea to use a point-to-point wiring diagram to review when making these checks.
- Apply low voltages or low power to check proper functionality of circuits.

HEAD OF THE DEPARTMENT

ANALOG ELECTRONIC CIRCUITS LAB

INDEX

S.NO	DATE	NAME OF THE EXPERIMENT	PAGE NUMBER	REMARKS

LIST OF EXPERIMENTS AR 20

1. Verify the process of Multiplexing and De-Multiplexing using TDM technique.
2. Study the characteristics of PCM transmitter and Receiver and also calculate the Bandwidth
3. Convert analog voltages into their digital equivalents using Delta Modulation and observe the waveforms.
4. Observe the process of Frequency ShiftKeying Modulation and Demodulation
5. Verify the process of Phase ShiftKeying Modulation and Demodulation.
6. Verify the operation of Differential PhaseShiftKeying.
7. Minimize coding redundancy using Source Encoder and Decoder
8. Perform error detection and correction using Linear Block Code-Encoder and Decoder
9. Perform error detection and correction using Binary Cyclic Code – Encoder and Decoder
10. Perform error detection and correction using Convolution Code – Encoder and Decoder

AUGMENTED EXPERIMENTS

1. Verify the process of Differential PulseCode Modulation and Demodulation.
2. Perform Non-Uniform Quantization using Companding.
3. Verify the operation of Differential Phase Shift Keying using MATLAB Simulink.
4. Observe the process of FrequencyShiftKeying Modulation and Demodulation using MATLAB Simulink.
5. Analyze the channel coding techniques in time and frequency domain.

BLOCK DIAGRAM : -

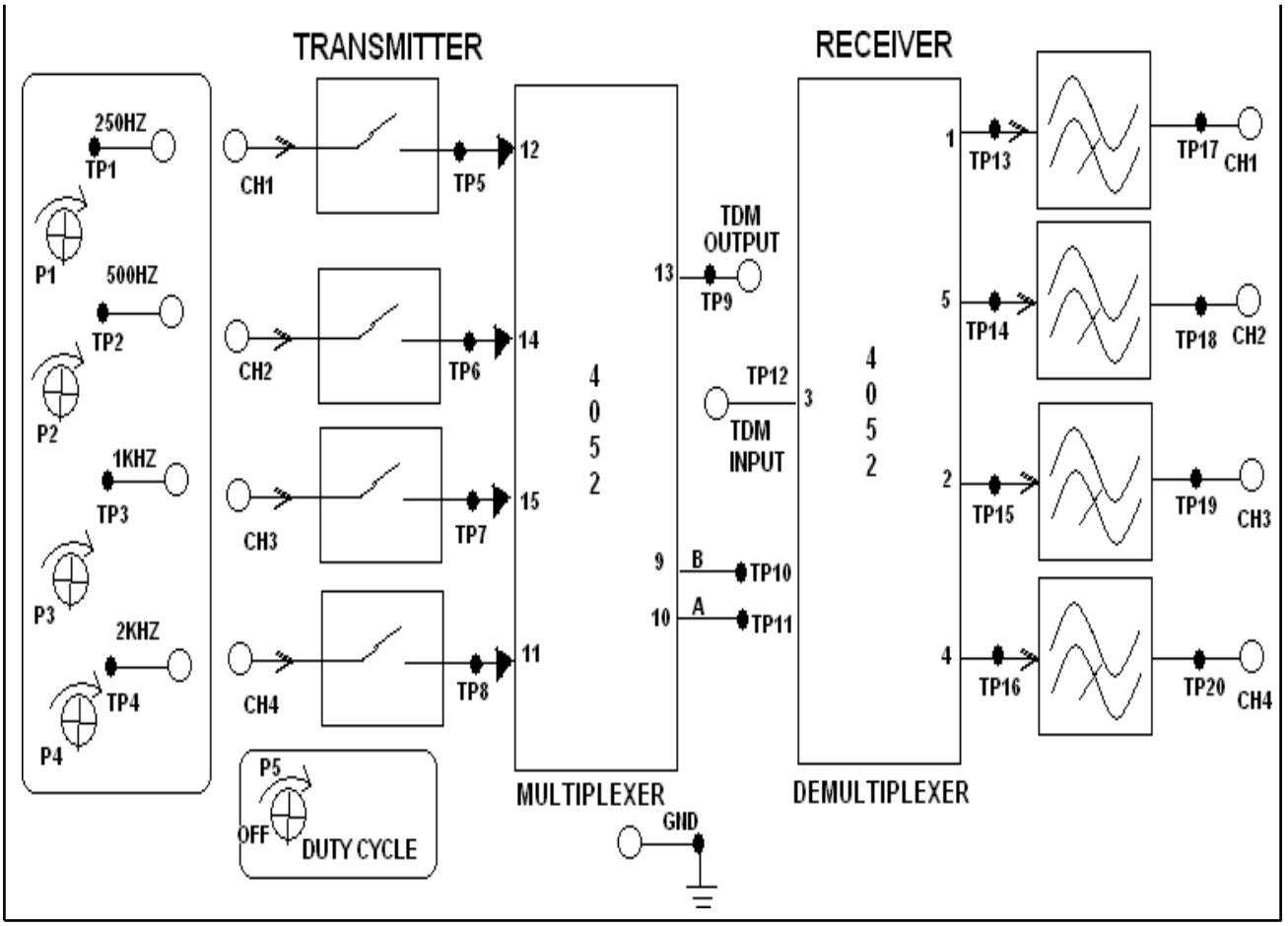


Fig: TIME DIVISION MULTIPLEXING & DEMULTIPLEXING

1. TIME DIVISION MULTIPLEXING & DEMULTIPLEXING

AIM: - 1.Study of 4 channel Analog Multiplexing and Demultiplexing Techniques.

2.Study of the effect of sampling frequency variation on the output.

3.Study of input signal amplitude on the output.

APPARATUS: -

1. TIME DIVISION MULTIPLEXING & DEMULTIPLEXING TRAINER.
2. Oscilloscope – 30 MHz Dual Channels.
3. Patch cards.

THEORY: -

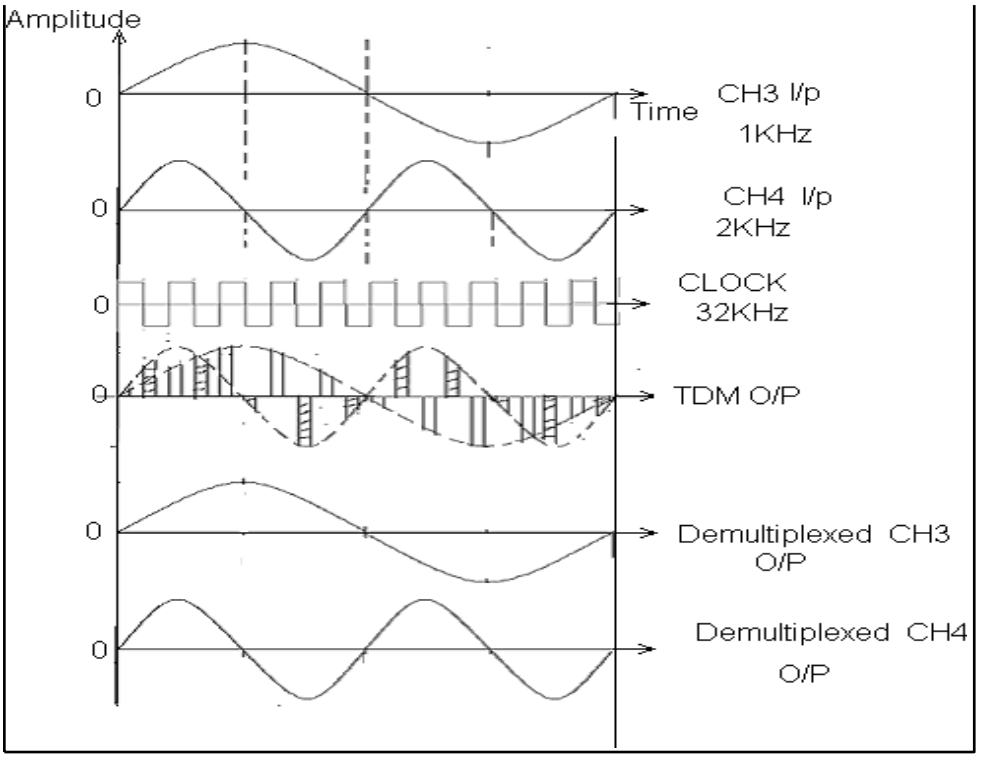
The TDM is used for transmitting several analog message signals over a communication channel by dividing the time frame into slots, one slot for each message signal. TheTDM, divides the available time into slots as shown in fig. The four input signals, all band limited by the input filters are sequentially sampled, the output of which is a PAM waveform containing samples of the input signals periodically interlaced in time. The samples from adjacent input message channels are separated by Ts/M, where M is the number of input channels is called a frame and is shown in the figure. The clock pulse generator output is a 20% duty cycle (for 4 message signals) square wave at 8 KHz. And an eight message system would require 10 % duty cycle and a 2.5 % quiet time between pulses.

PROCEDURE:

MULTIPLEXING

1. Connect the circuit as shown in diagram.
2. Switch ON the power supply.
3. Set the amplitude of each modulating signal as 5V peak to peak.
4. Monitor the outputs at test points 5,6,7,8. These are sampling PAM outputs..
5. Observe the outputs by varying the duty cycle pot (P5). The PAM outputs will vary with 10% to 50% duty cycle.
6. Try varying the amplitude of modulating signal corresponding each channel by using amplitude pots P1, P2, P3, P4. Observe the effect on all outputs.
7. Observe the TDM output at pin no. 13 (at TP9) of 4052, all the multiplexed channels are observed during the full period of the clock (1/32 KHz)

MODEL GRAPH : -



OBSERVATIONS : -

	Amplitude	Time Period	Frequency
Channel 3			
Channel 4			
Demultiplexed CH3			
Demultiplexed CH 4			
Clock			
TDM O/P CH3			
TDM O/P CH4			

T_{ON}= 15µs T_{OFF}=15µs

DEMULTIPLEXING & LOW PASS FILTER

1. Connect the circuit as shown in diagram.
2. Observe the demultiplexed outputs at test points 13.14,15,16 respectively.
3. Observe by varying the duty cycle pot P5 and see the effect on the outputs.
4. Observe the low pass filter outputs for each channel at test points 17,18,19,20 and at sockets channels CH1, CH2, CH3, CH4. These signals are true replica of the inputs.
5. These signals have lower amplitude.

RESULT : -

Hence the time division multiplexing and demultiplexing techniques are performed and the effect of sampling frequency variance on output is studied and corresponding graphs are plotted.

BLOCK DIGRAM

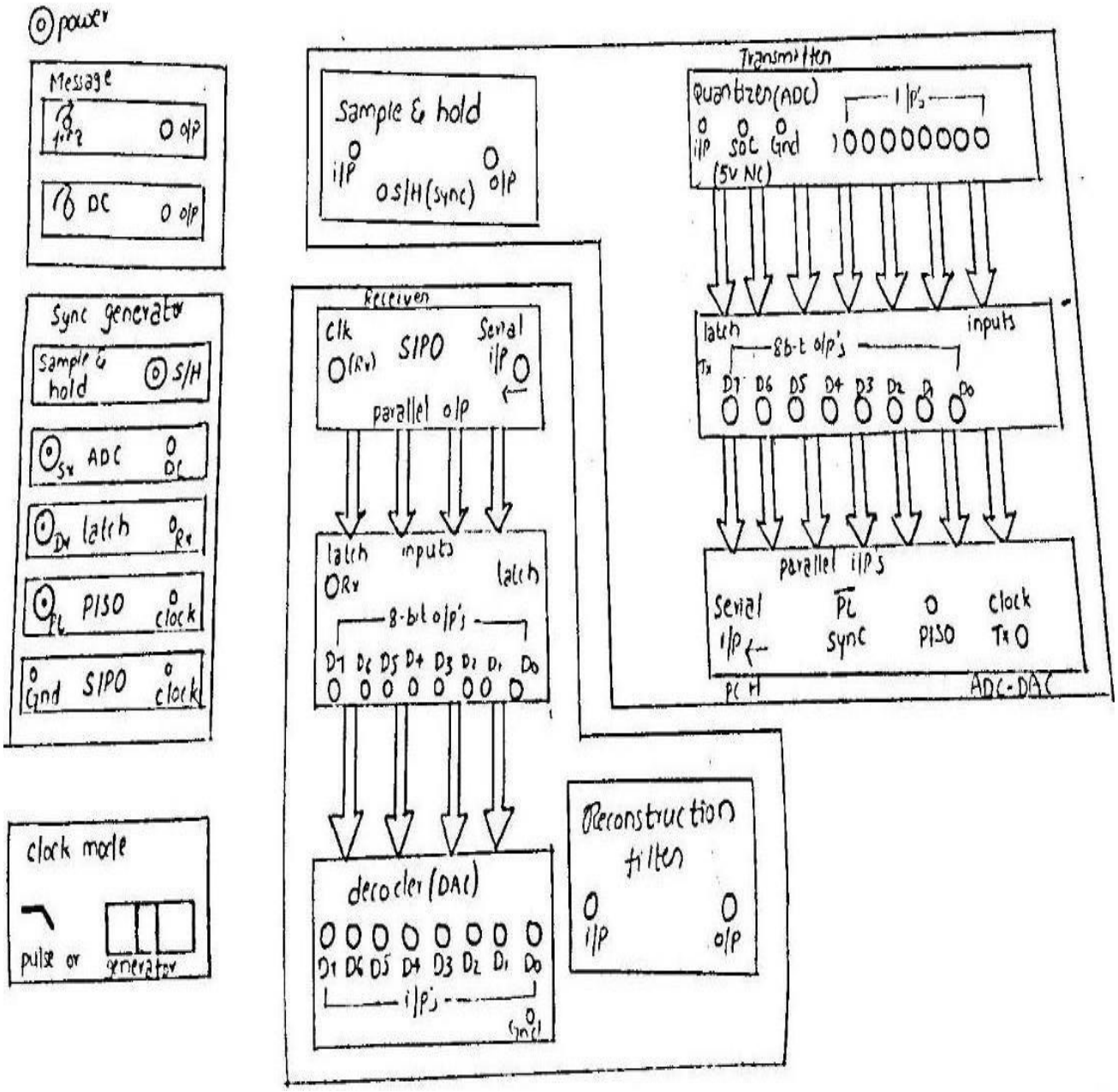


Fig: PULSE CODE MODULATION & DEMODULATION

2. PULSE CODE MODULATION & DEMODULATION

AIM: -

1. To study the pulse code modulation and demodulation.

2. To study the effect of A-Law PCM and Linear modulation by varying the input signal amplitude.

APPARATUS:

1. PCM Modulation & Demodulation Trainer Kit.
2. 30 MHz Dual Channel CRO.

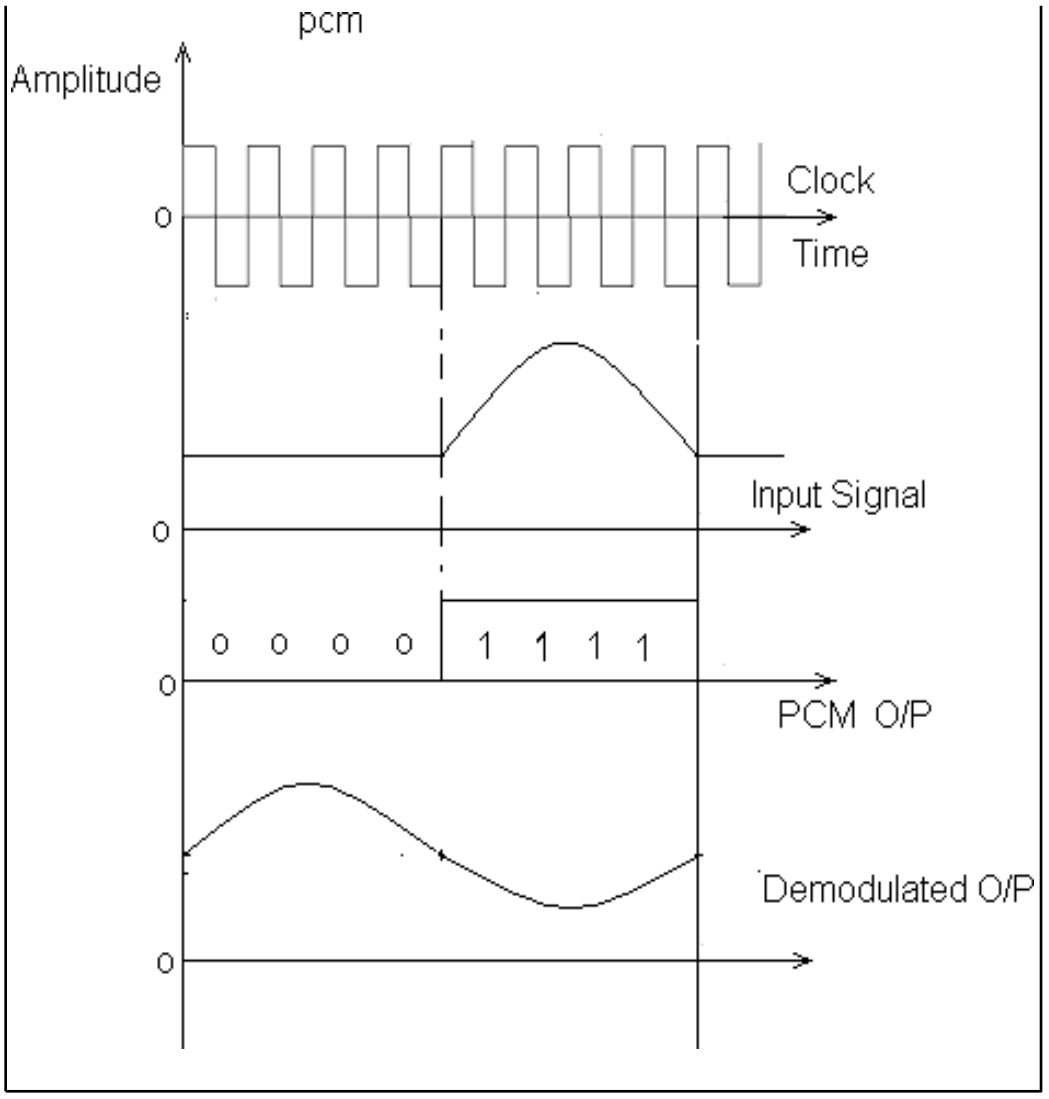
THEORY:

In analog communication systems, the limitation is that once noise is introduced at any place along the channel, we are 'stuck' with it. To overcome such a situation, a process known as 'Quantization' is introduced in which the analog signal is approximated to the nearest whole number of small steps of each size
S. This process of quantization is done during every sampling interval. Sampling of the Analog signal however is done at a rate keeping in view of the Nyquist criterion.

In the process of quantization, approximation is made to make the number of steps a whole number resulting in an approximation error of ± S/2. The approximation noise or quantization noise is reproduced in the demodulation and is inseparable from the original signal.

The quantized steps represented a binary number (of pulses) is transmitted to the receiver, this whole process of sampling, quantization and conversion into Binary code is known as Pulse code Modulation (PCM). Thus in PCM the code represented at a particular time slot (sampling period) is representative of the instantaneous amplitude of the Analog signal.

MODEL GRAPH:


pcm
Amplitude
Clock
0
Time
Input Signal
0
0 0 0 0 1 1 1 1
0
PCM O/P
Demodulated O/P
0

PROCEDURE: 1.To study the Pulse Code Modulation and Demodulation.

Switch on the power supply.Connect the CRO probe at TP2 and set CRO in mode then vary the pot P2, observe the varable DC voltage (-2.5v to +2.5v).

1. Connect the patch cord from TP2 to MOD. Input. Observe the level shifter output at TP7, which is shifted by +2.5v
2. Keep pot P2 –2.5v side (fully anticlock wise), now observe all LED's OFF. Now turn P2 slowly clock wise and observe LED's glowing. Observe the serial data at TP9.
3. Connect CRO channel 1 probe at TP2 and Channel 2 probe at Demod. Output TP21. Vary pot P2; observe that same DC voltage should appear at two points.
4. Keep some DC voltage, select one type of Encoding method by selecting toggle switch (A-Law or Linear) then observe LED's. Identify the bit pattern between Linear and A- Law modulation. The 8-bit compressed A-Law values for 13-bit input values given in table no.
5. Remove DC from input and connect 1KHz sine wave at TP1.
6. Observe sampling rate clock at TP8 by changing sampling rate at toggle switch (4KHz and 8KHz).

Observe LPF output at TP6, level shifter output at TP7.

7. Observe the Sampling clock at TP8 by changing SAMPLING RATE switch and also observe A/D QUANTIZER output at TP9 (Serial data output).
8. Observe the Demodulated output at TP21. Identify the difference between output waveforms when sampling frequency is 4KHz and 8 KHz.
9. By changing Amplitude of sine wave observe Demodulatedoutputs. Observe the DC voltage at TP7, it is +2.5v. This is origin for ADC (single supply). Any input signal applied at MOD. Input shifted to this voltage level. This voltage is indicating on output LED's when no input is applied.

OBSERVATIONS:

For DC input:

Linear PCM: -

S.No.	DC I/P	Level shifter O/P	PCM O/P TP10 TP11 TP12 TP13 TP14 TP15 TP16 TP17	Receiver O/p
1				
2				
3				
4				
5				
6				
7				

A- Law PCM:

S.No.	DC I/P	Level shifter O/P	PCM O/P TP10 TP11 TP12 TP13 TP14 TP15 TP16 TP17	Receiver O/p
1				
2				
3				
4				
5				
6				
7				

2. To Study the effect of A-Law in PCM:

1. Connect Mod. Input from external input at TP4. Connect external sine wave from function generator at TP4. Maximum amplitude of this waveform is $5V_{p-p}$.
2. Connect CRO ch1 at TP4 and ch2 at TP21. Set input amplitude $200mV_{p-p}$ in function generator. Keep sampling rate switch in 8 KHz and keep encoding switch in A-Law.
3. Observe Demodulated utput at TP21. Now change encoding switch to Linear, observe Demodulated output at TP21. Identify deference between A-law PCM and linear modulation. In linear modulation Demodulated output is not same as input in shape. In A-Law PCM output is same as input in shape. Repeat this process for decreasing input amplitude and varying frequency.

RESULT:

Hence the Pluse code modulation and demodulation techniques are observed.

BLOCK DIAGRAM

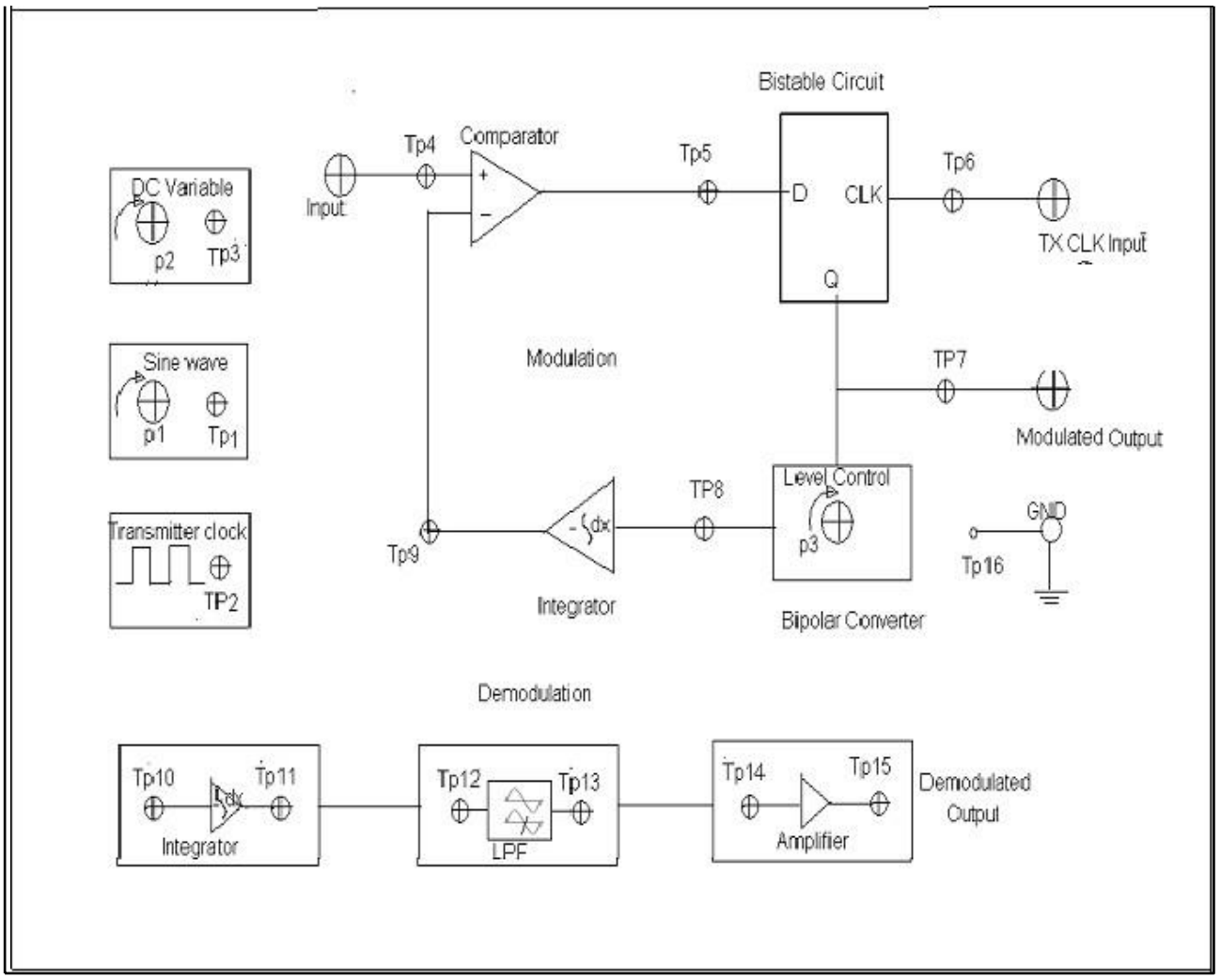


Fig: DELTA MODULATION & DEMODULATION

3. DELTA MODULATION & DEMODULATION

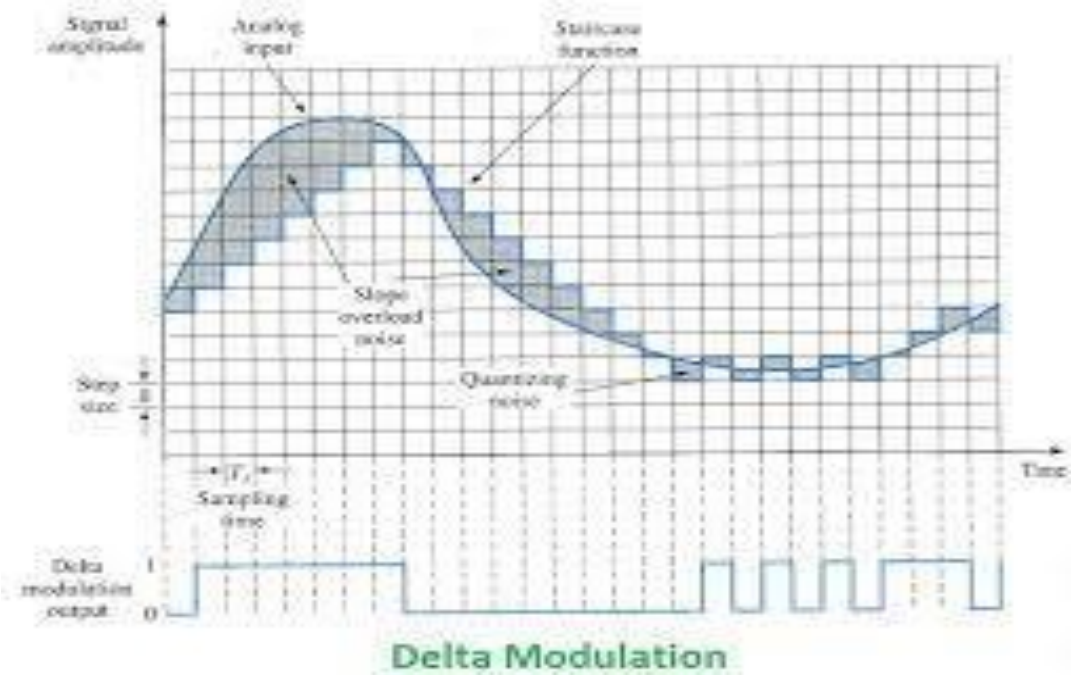
AIM: -

To study the Delta Modulation and Demodulation.**APPARATUS: -** 1. Delta Modulation & Demodulation Trainer Kit. 2. Oscilloscope – 30 MHzdual channels. 3. Patch Cards

THEORY:

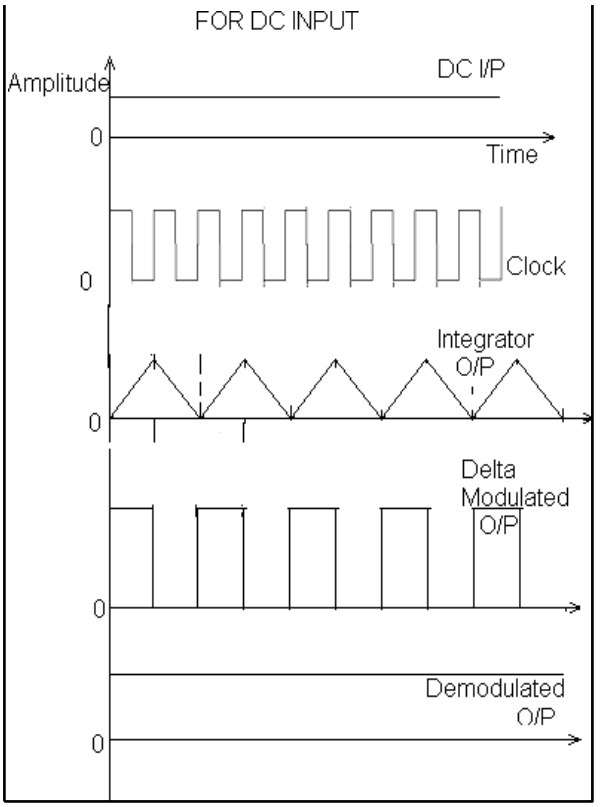
In Delta Modulation, the incoming signal is over sampled (much higher than Nyquist rate) to purposefully increase the correlation between adjacent samples of the signal. In DM, a staircase approximation to the over sampled version of the message signal is shown in fig.

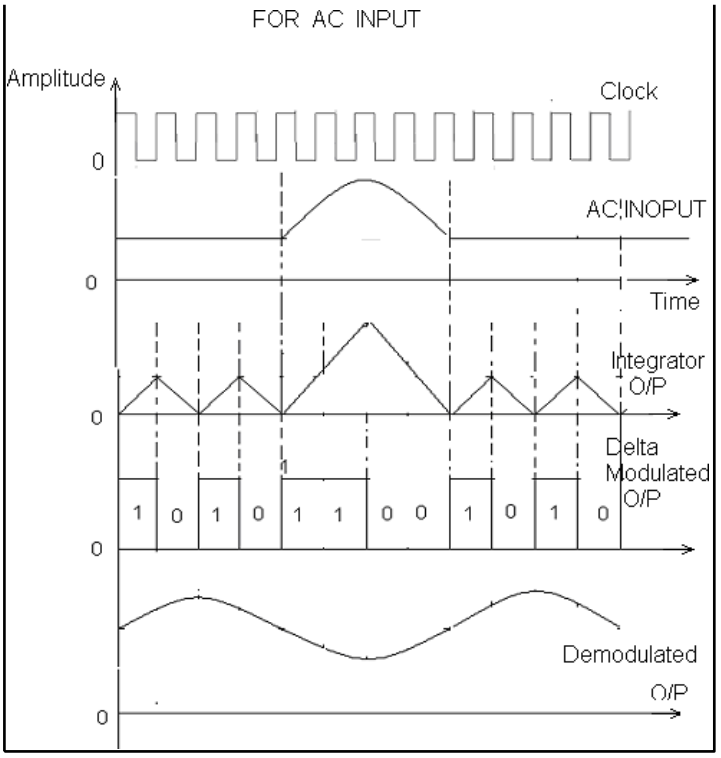
The difference between the input and the approximation is quantized into only two levels (± Δ) corresponding to positive and negative differences respectively. Thus if the approximation falls below the signal at any sampling epoch, it is increased by Δ are if approximation lies above the signal, it is diminished by Δ . Provided the signal does not change too rapidly from sample to sample, the approximation remains within ±Δ of the input signal.



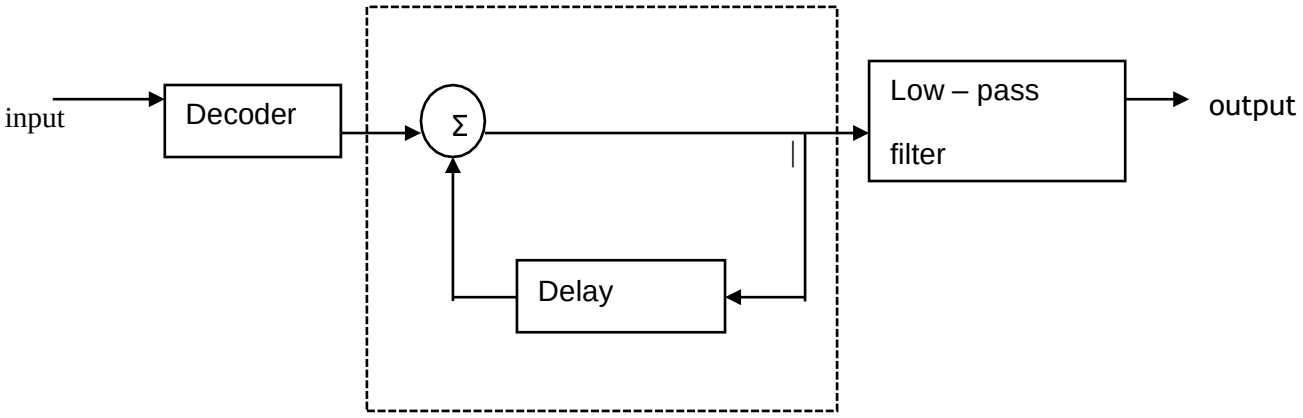
Delta Modulation

MODEL GRAPH:


FOR DC INPUT
Amplitude
DC I/P
0
Time
0
Clock
Integrator
O/P
0
Delta
Modulated
O/P
0
Demodulated
O/P
0


FOR AC INPUT
Amplitude
Clock
0
AC INOPUT
0
Time
Integrator
O/P
0
Delta
Modulated
O/P
1 0 1 0 1 1 0 0 1 0 1 0
0
Demodulated
O/P
0

In DM system the rate of information transmission simply equal to the Sampling rate, multiplied by No of bits per sample.The DM may be generated by applying the sampled version of the incoming message to modulator that involves a comparator, quantizer and accumulator interconnected as shown in fig.



Accumulator

The comparator computes the difference between its two inputs. The quantizer consists of a hard limiter with an input – output relation, which is a scaled version of signum function.

The quantizer output is then applied to an accumulator to produce.

$$m_q(nTs) = \sum_{i=1}^{n} e_q(its)$$

Thus at the sampling instant nTs, the accumulator incremented the approximation by a step Δ in a positive or negative direction depending on the algebraic sum sign of the error signal e(nTs).

In the receiver shown in fig the staircase approximation mq(t) is reconstructed by passing the sequence of positive or negative pulses , produced at the decoder output ,through an accumulator in a manner similar to that used in the transmitter . the out of land quantization noise in the high frequency staircase waveform mq(t) is rejected by passing it through a low pass filter , with a band with equal to the original message bandwidth.

OBSERVATIONS:
For AC Input:

	Amplitude	Time Period	Frequency
DC Input			
AC Signal			
INTEGRATOR O/P			
DELTAMODULATEDO/P			
DEMODULATED O/P			

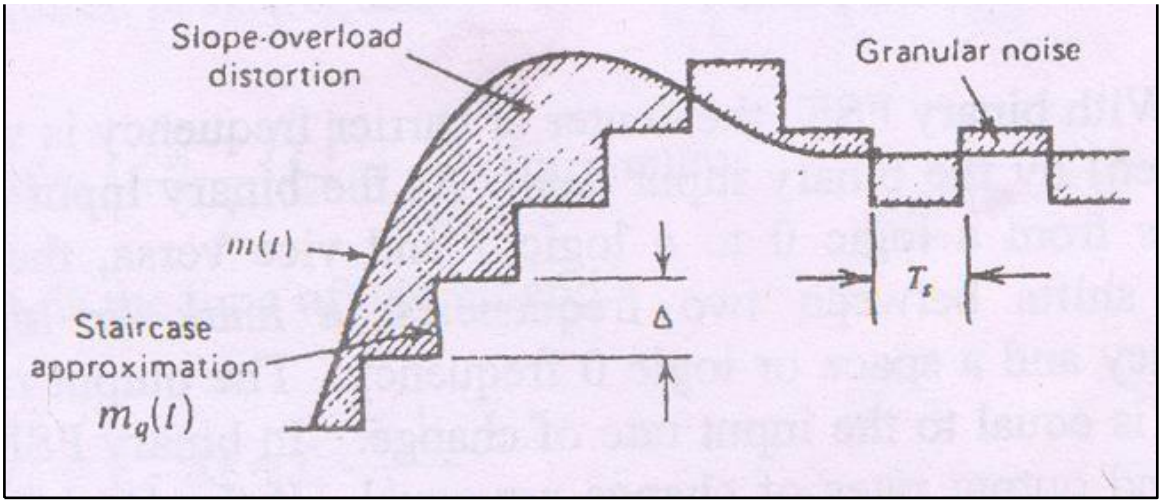
For DC Input:

	Amplitude	Time period	Frequency
DC Input			
TXN Clock			
INTEGRATOR O/P			
DELTA MODULATED O/P			
DEMODULATED O/P			

The DM is subjected to two types of quantization error,

1) Slope overload distortion and 2) Granular Noise. The maximum slope of the staircase approximation $m_q(t)$ is fixed by the step size Δ, increases and decreases in $m_q(t)$ tend to occur along straight lines. A delta modulator using a fixed step size is referred to as a Linear Delta Modulator. The Granular noise occurs when the step size Δ is too large relative to the local slope characteristics of the input waveform m(t) causing the stair case approximation $m_q(t)$ to hunt around a relatively flat segment of the input waveform as shown in fig.

This granular noise is similar to quantization noise in PCM system.



Thus a large step size is required for accurate representation of relatively low – level signals. As a compromise we need to make the delta modulator "ADAPTIVE", that is the step size is made to vary in accordance with the input signal.

PROCEDURE: -

1. Switch ON the power supply.
2. Connect TX clock from TP2 to TX clock input TP6. The clock frequency should be 32KHz.
3. In order to ensure for correct operation of the system, we first take the input to 0v from DC variable at TP3 so connect the '+'input of the delta modulator's VOLTAGE COMPARATOR to 0v.
   Observe the output of integrator at TP9 and the output of the level change the TP8
   If the Transmitter's LEVEL CHANGER output has equal positive and negative output levels, Integrator's output will be a triangle wave centered around '0'volts,as shown in fig. However, if the level changer's negative level is greater than the positive level, the integrator's output will appear as shown in fig. Should the level changer's positive output level be the greater of the two levels, the integrator's output will resemble that shown in fig.

The relative amplitudes of the level changer's positive and negative output levels can be varied by adjusting the LEVEL ADJUST present in the BISTABLE AND LEVEL CHANGER CIRCUIT.

4. The output from the Transmitter's BISTABLE circuit (tp7) will now be a stream of alternate '. 1'and '0's,this is also the output of the delta modulator itself. The delta modulator is now said to be balanced for correct operation.
5. Disconnect the voltage comparator's '+' input from 0V, and reconnect it to the 2KHz sine wave from TP1.
6. Observe the integrator output by varying the amplitude.
7. Observe the Bistable output together with the analog input at TP4, and note that the 2KHz sine wave has effectively been encoded into a stream of data bits at the Bistable's output.
8. Connect the modulation output to the integrator input at TP10 and observe the output at TP11.
9. Now connect TP11 to the low pass filter input TP12 and observe the output.
10. Connect TP13 to the amplifier input TP14 and observe the demodulation output at TP15.

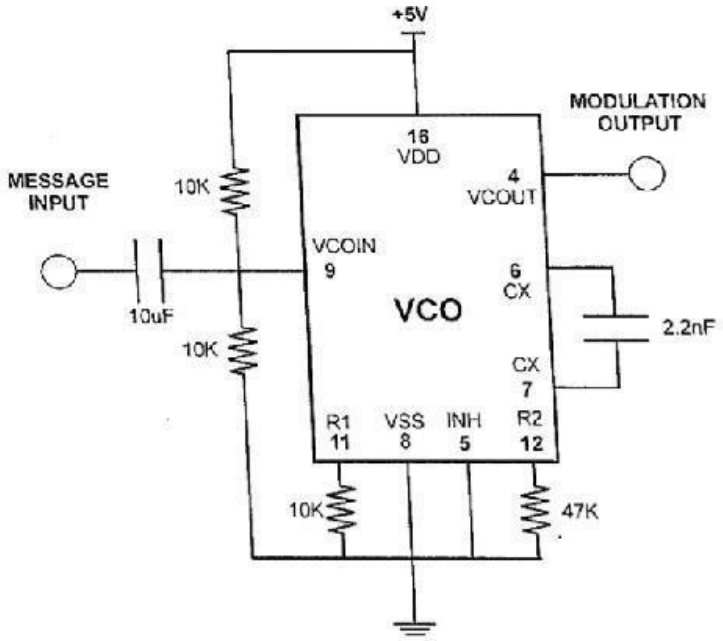
RESULT:

Delta modulation and demodulation is performed and there output waveforms are plotted.

.

BLOCK DIGRAM

FSK Transmitter



FSK Receiver

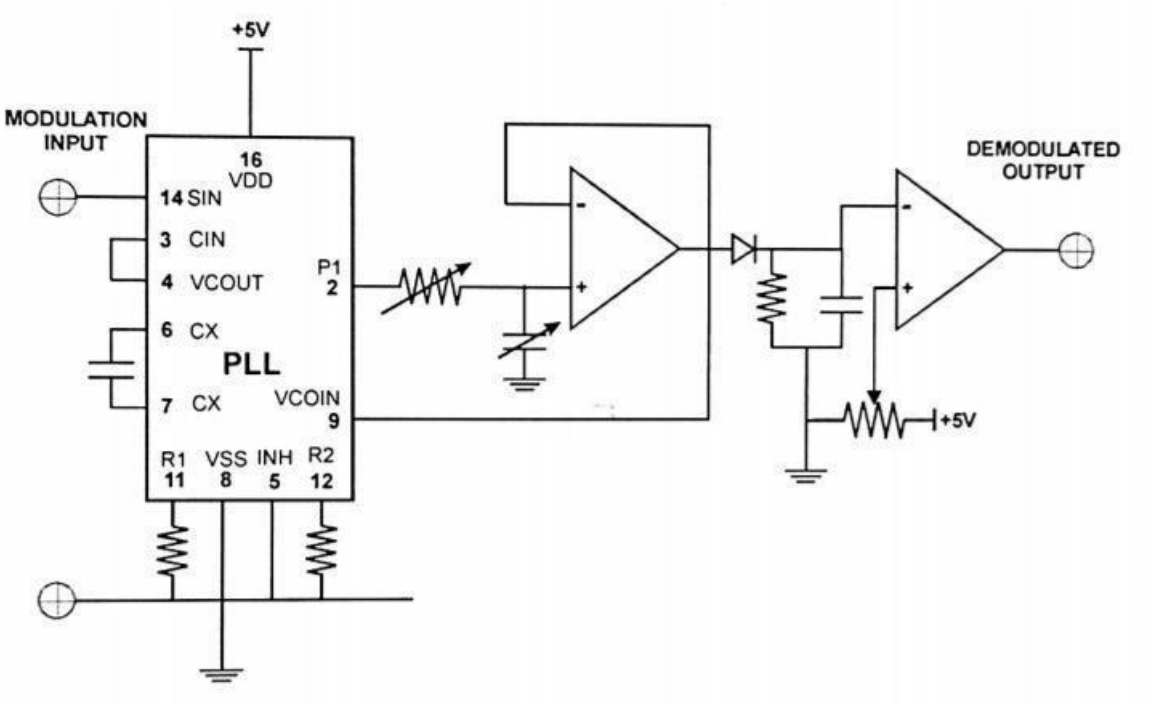


Fig: FREQUENCY SHIFT KEYING MODULATION & DEMODULATION

4. FREQUENCY SHIFT KEYING MODULATION & DEMODULATION

AIM: -

1. To generate FSK Modulation.
2. To Demodulate the FSK signals,
3. To generate (NRZL), RZ, NRZ (M), BIPHASE (MARK), BIPHASE (MANCHESTER).

APPARATUS:-

1. Fsk Modulation & Demodulation Trainer Kit.
2. C.R.O.30 MHz Dual Channel.
3. Patch Cards.

THEORY:

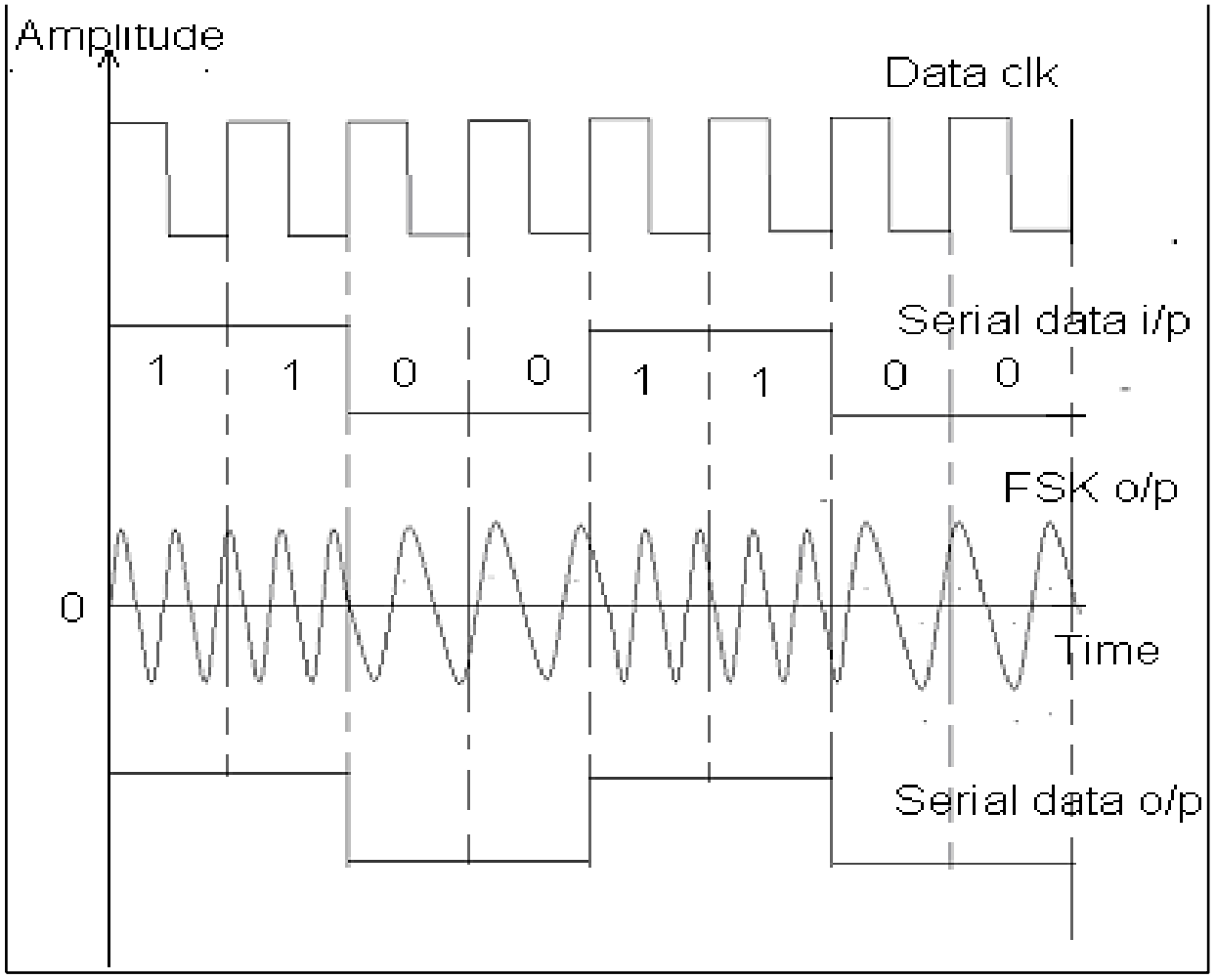
Binary FSK is a form of constant – amplitude angle modulation and the modulating signal is a binary pulse stream that varies between two discrete voltage levels but not continuous changing analog signal. In FSK, the carrier amplitude (Vc) remains constant with modulation and the carrier radian frequency (Wc) shifts by an amount equal to $\pm \Delta W/2$. The frequency shift ($\Delta W/2$) is proportional to the amplitude and polarity of the input binary signal. For example, a binary 1 could be +1 volt and a binary zero could be -1 volt producing frequency shifts of $+ \Delta W/2$ and $- \Delta W/2$ respectively. The rate at which the carrier frequency shifts is equal to the rate of change of the binary input signal Vm (t) (that is the input bit rate). Thus the output carrier frequency deviates (shifts) between $Wc + \Delta W/2$ and $Wc - \Delta W/2$ at the rate equal to fm.

FSK TRANSMITTER:

With binary FSK, the center or carrier frequency is shifted (deviated) by the binary input data .As the binary input signal changes from a logic 0to a logic 1 and vice versa, the FSK output shifts between two frequencies : a mark or logic 1 frequency and a space or logic 0 frequency . The output rate of change is equal to the input rate of change. In the binary FSK, the input and output rates of changes are equal. If the fundamental frequency of the input is considered, the highest modulating frequency is equal to one – half the input bit rate.

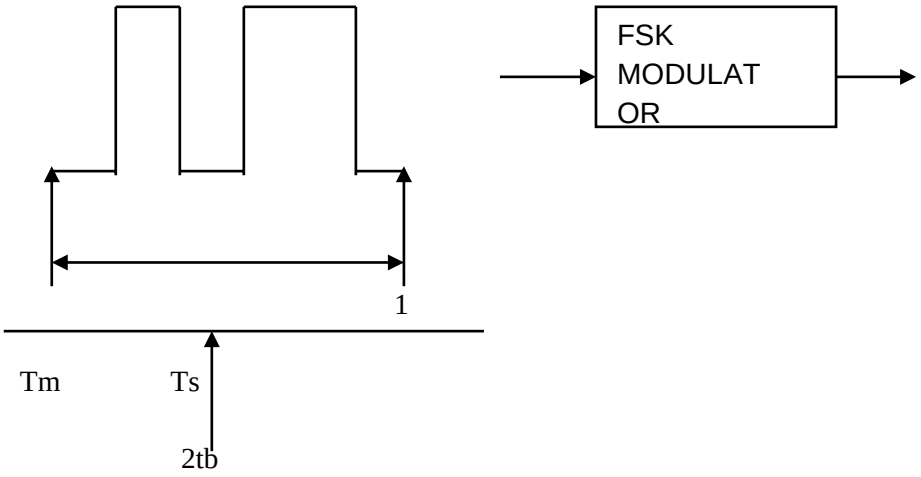
The frequency of the VCO is chosen so that falls half way between the mark and space frequencies. In binary FSK, Δ f is the peak frequency deviation of the carrier and is equal to the difference between the rest frequency and either mark or space frequency. In a binary digit signal , the frequency deviation is constant and always at its maximum value , as logic is have the same voltage and all logic 0s have the same voltage .

MODEL GRAPH:


Amplitude
Data clk
Serial data i/p
1
1
0
0
1
1
0
0
FSK o/p
0
Time
Serial data o/p

The output of FSK modulator is related to binary input , a logic 0 corresponds to space frequency fs and a logic 1 corresponds to mark frequency fm and fc is thee carrier the frequency

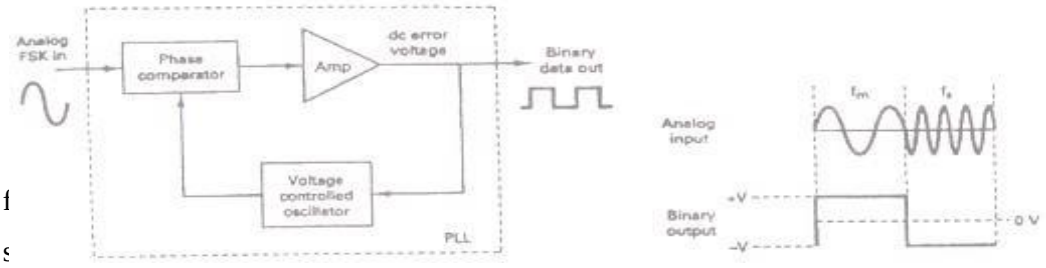
1 0 1 1



The required peak frequency deviation Δf

$\Delta F = (|fm - fs|)/ 2 = 1 / 4\ tb$ (minimum)

Where tb is the time of one bit and $fm = fc - \Delta f = fc - (1/4tb)$



f

s

$= fc - \Delta f = fc + (1/4tb)$

FSK thus consists of two pulsed sinusoidal waves of frequency fm and fs.

The bandwidth for FSK = BW = fm – fs +(2/tb)

OBSERVATIONS

	Amplitude	Time Period	Frequency
Message Signal			
Carrier Signal (VCO Output without message signal)			
FSK Output For Logic 1 For Logic 0			
Frequency Shift			
Demodulated Output			

FSK RECEIVER:

As shown in fig, the incoming FSK signal is multiplied by a recovered carrier signal which has same frequency and phase as the transmitter carrier (the two transmitted frequencies – mark and space are not continuous. Due to practical difficulty in reproduction of a local reference, this detection is not used).

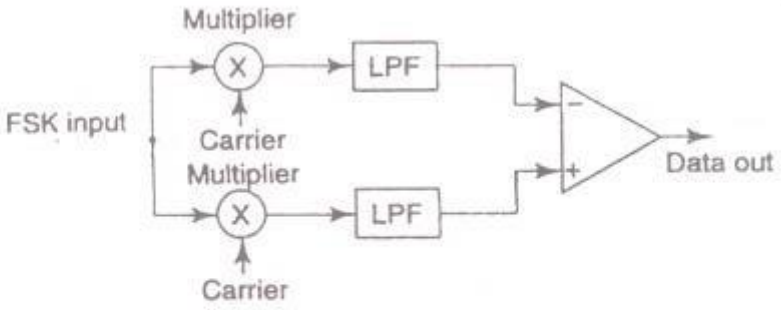


Fig : Coherent FSK Demodulator

The most common circuit for demodulating binary FSK signals is the phase – Locked Loop (PLL) as shown in fig.

As the input to the PLL shifts between the mark and space frequencies, the D.C. error voltage at the output of the phase comparator follows the frequency shift. As there are only two input frequencies (mark and space), there are also only two output error voltage. One represents a logic 1 and the other a logic 0. Therefore the output is a two level (binary) representation of the FSK input. Generally the natural frequency of the PLL is made equal to the center frequency of the FSK modulator. As result the changes in the D.C error voltage follow the changes in the analog input frequency and are symmetrical around 0 volts.

Binary FSK has a poorer error performance than PSK or QAM and is not used for high performance digital radio systems and its use is restricted to low performance, low cost and asynchronous data modems.

PROCEDURE: -Modulation: -

1. Switch ON the power supply.
2. Set the Data selection switch (DATA SELECTION) to the desired code (say11001100).
3. Set the switch (DATA ON-OFF) ON position. Observe the 8 Bit Word pattern at TP12.
4. Observe the Data Clock at TP1 and also observe the NRZ (L) at TP2, RZ at TP3, NRZ9M) at TP4, BIPHASE (MARK) at TP5, BIPHASE (MANCHESTER) at TP6.
5. Connect the patch cord as shown in diagram1. Observe the corresponding FSK output at (when Data is logic '1', the frequency is high and Data is logic "0" the frequency is low) TP8.
6. Repeat the step 5 for other inputs. (Like NRZ (M), RZ, BIPHASE). Observe the corresponding FSK output
7. Now change the Data selection and repeat the above steps 3 to 6 and observe the corresponding FSK outputs.

DEMODULATION: -

1. Connect the patch cords as shown in diagram.
2. The incoming FSK input is observed at TP9.
3. The output of 'Square wave converter' is available at TP10. The serial Data output is a available at TP11.
4. Repeat the above steps 1,2,3 for other serial data outputs. The outputs are truw replica of the original inputs.

RESULT:

Hence the modulation and demodulation of FSK are generated.

BLOCK DIAGRAM:

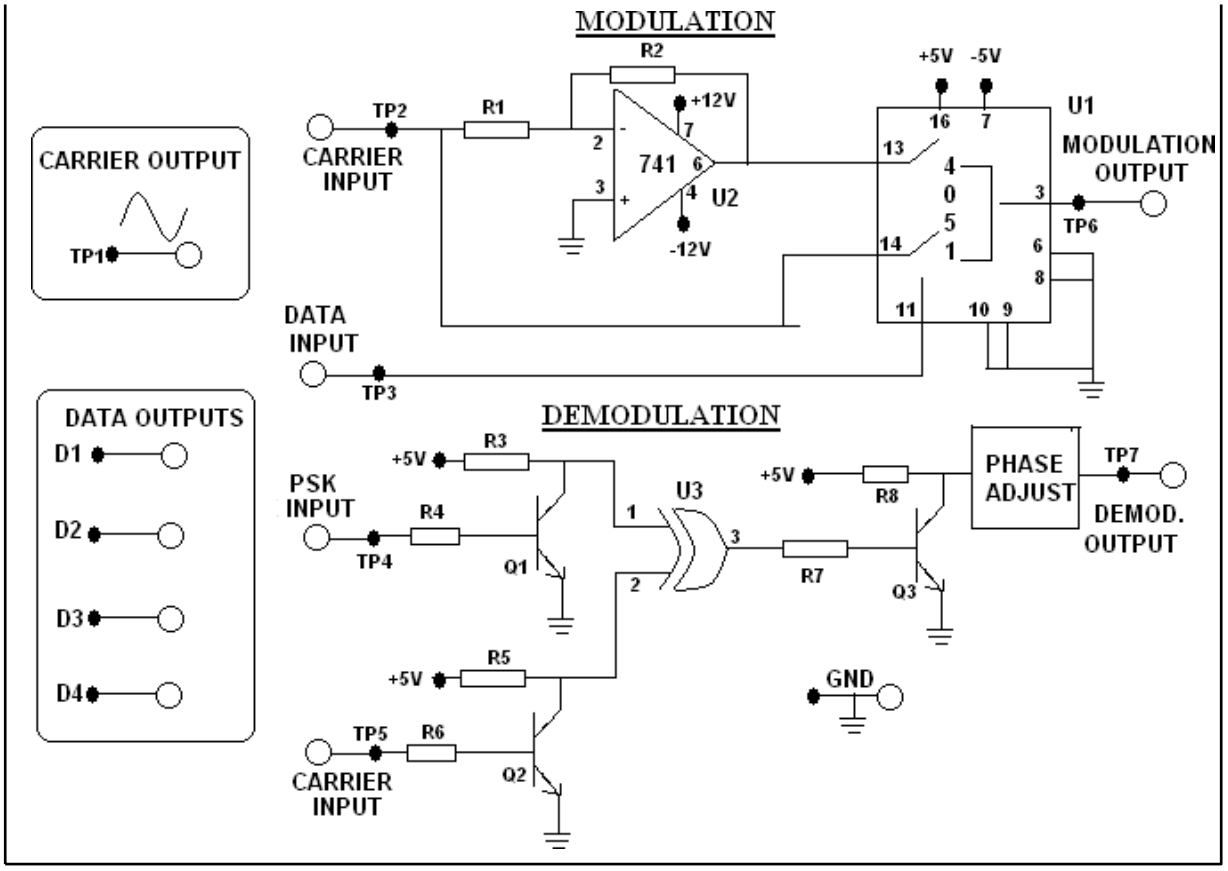


Fig: PHASE SHIFT KEYING MODULATION & DEMODULATION

5. PHASE SHIFT KEYING MODULATION & DEMODULATION

AIM: -

T study the operation of PHASE SHIFT KEYING modulation and demodulation Techniques.

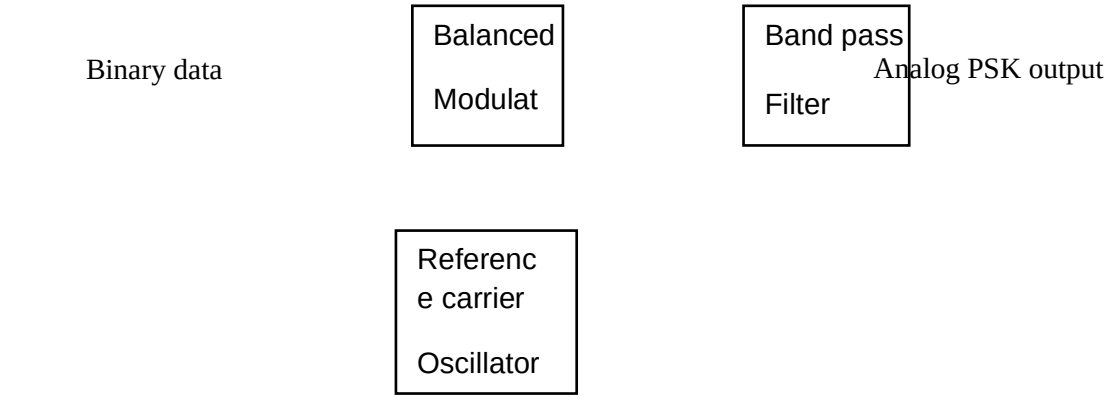
APPARATUS:

1. Psk Modulation and Demodulation Trainer Kit.
2. Oscilloscope 30 MHz, Dual Channel.
3. Patch chords.

THEORY:

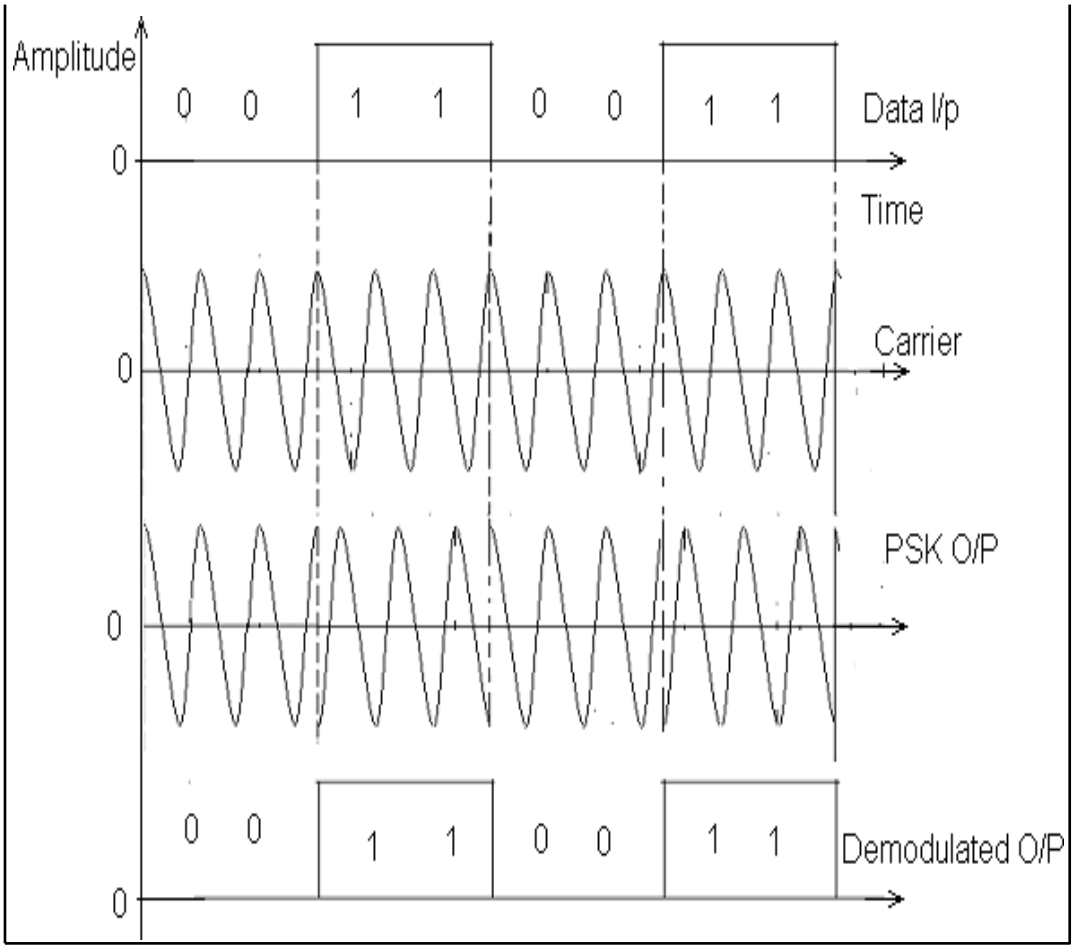
The PSK is a form of angle modulated, constant amplitude digital modulation. In BPSK, two output phases are possible for a single carrier frequency. As the input digital signal changes state, the phase of the output carrier shifts between 180º out of phase.

As shown in BPSK modulator in fig A and B the balanced modulator acts as a phase reversing switch. Depending on the logic condition of the digital input, the carrier is transferred to the output either in phase or 180º out of phase with reference carrier oscillators and for proper operation, the digital input voltage must be much greater than the peak carrier voltage as it has to control ON – OFF state of diodes



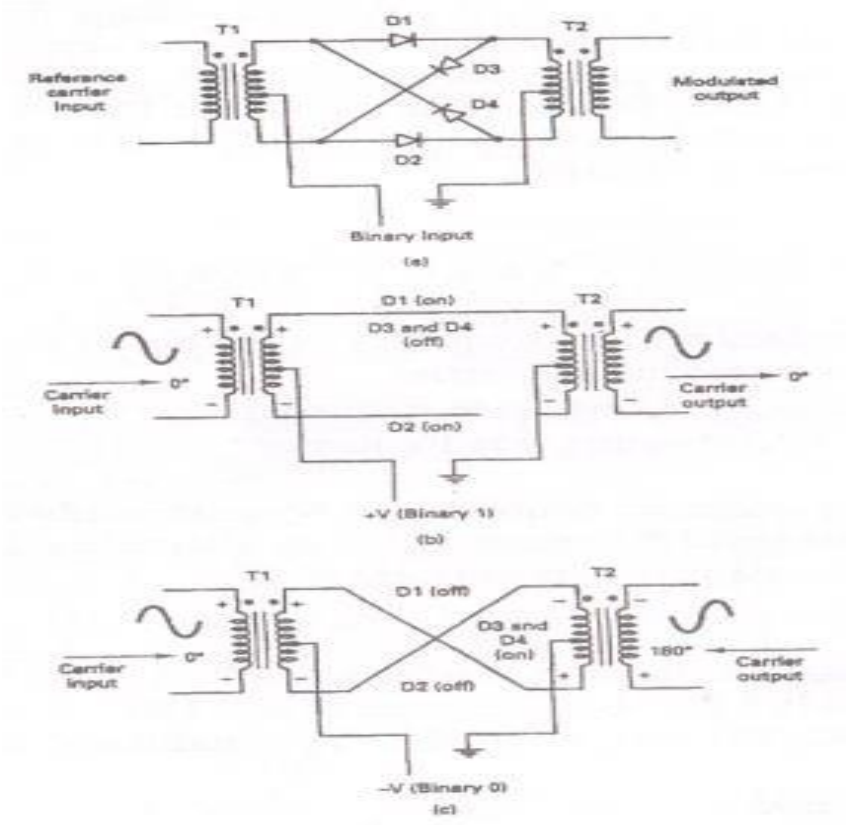
A) BPSK Modulator

MODEL GRAPH:


Amplitude
0 0 1 1 0 0 1 1
Data I/p
0
Time
Carrier
0
PSK O/P
0
0 0 1 1 0 0 1 1
Demodulated O/P
0

OBSERVATIONS: -

	Amplitude	Time Period	Frequency	Phase
Carrier Input				
Data Input Options: D1 D2 D3 D4				
PSK Output For Logic 1 For Logic 0				
Demodulated Output				



PROCEDURE:

1. SwitchON the trainer and see that the supply LED glows.
2. Observe the carrier output at TP1.
3. Observe the data outputs (D1, D2, D3, D4).
4. Now, connect the carrier output TP1 to the carrier input of PSK modulator TP2 using patch chord.
5. Connect the D1 to data input of PSK modulator TP3.
6. Observe the Phase shifted PSK output waveform on CRO on channel1 and corresponding data output on channel 2.
7. Repeat the steps 4,5,6 for data outputs D2, D3, and D4 and observe the PSK outputs.
8. Connect the carrier output TP1 to the PSK input of demodulation TP4.
9. Connect the carrier output TP1 to the carrier input of PSK demodulation TP5.
10. Now, observe the PSK demodulated output at TP7 on CRO at channel1 and corresponding data output on channel2.
11. The demodulated output is true replica of data output.
12. Repeat the steps 8 to 10 for other data outputs D2, D3, D4.

RESULT:

The operation of PSK modulation and demodulation techniques are studied.

BLOCK DIAGRAM:-

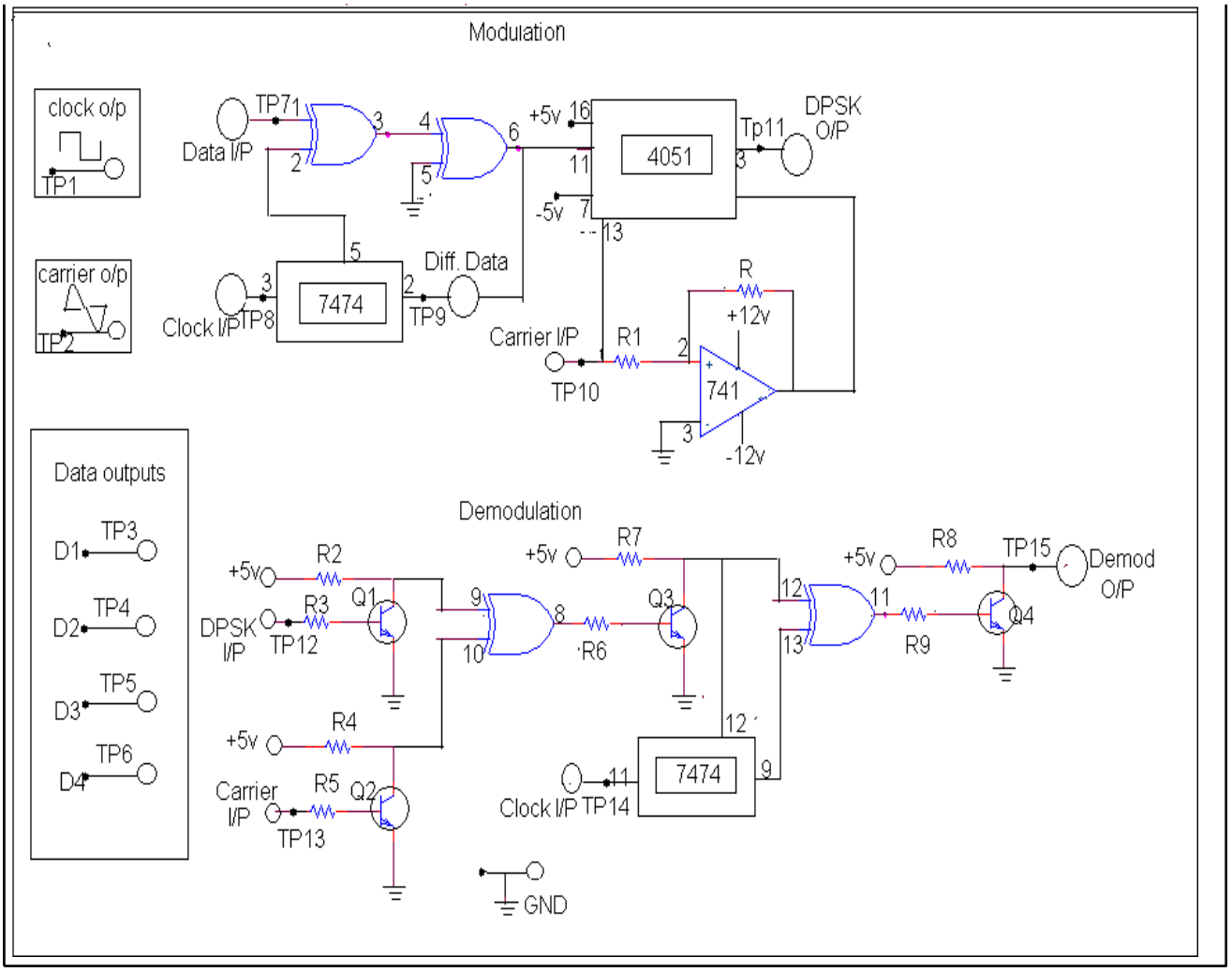


Fig: DIFFERENTIAL PHASE SHIFT KEYING MODULATION & DEMODULATION

6. DIFFERENTIAL PHASE SHIFT KEYING MODULATION & DEMODULATION

AIM : -

To study the operation of Differential Phase Shift Keying modulation & demodulation Techniques & plot the corresponding waveforms.

EQUIPMENT: -

1. Hi-Q Electronics, Dpsk Modulation & Demodulation Trainer.
2. Oscilloscope 30MHz,Dual Channel.
3. Patch chords.

THEORY : -

The DPSK is a non-coherent version of PSK. In coherent detection, the carrier wave's phase reference should be known for obtaining optimum error performance. The DPSK eliminates the need for a coherent reference signal at the receiver by combining two basic operations at the transmitter:

1) Differential Encoding of the input binary wave
2) Phase-shift keying

And hence the name differential phase shift keying. Thus to send and hence the name differential phase advance the current signal waveform by 180^o and to send 1, we have the phase of the current signal waveform unchanged. The receiver is equipped with a storage capability so that it can measure the relative phase difference between the waves forms received during two successive bit intervals. Provided that the unknown phase θ contained in the received wave varies slowly (slow enough and considered essentially constant over two bit intervals), the phase difference between waveforms received in two successive bit intervals will be independent of θ.

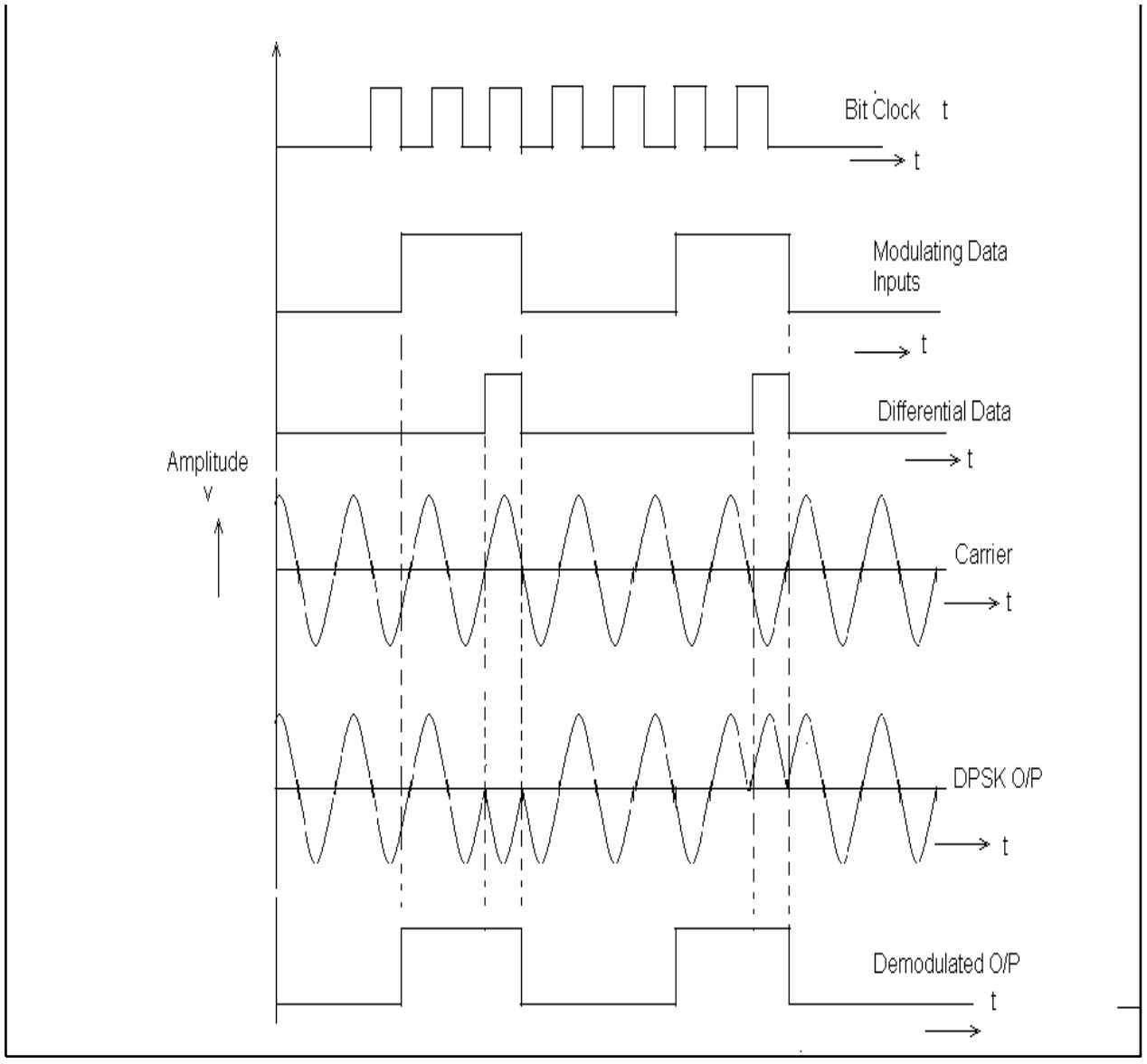
If the phase shift versus frequency is linear, delay is constant with frequency. If all frequencies are not delayed by the same amount of time , the phase frequency relationship is not linear and the received signal is distorted. The DPSK is an example of non – coherent orthogonal modulation when it is considered over two bit intervals.

The average probability of error or bit error rate for DPSK provide 3 – db over non – coherent FSK for the same Eb/No.

OBSERVATIONS: -

	Amplitude	Time Period	Frequency	Phase
1.Clock Input				
2. Carrier				
3.Differential Data				
4.Data Input Options: D1 D2 D3 D4				
5.DPSK O/P Logic – 1: Logic – 0:				
6. Demodulated O/P				

MODEL WAVEFORMS : -



PROCEDURE: -

1. Now switch ON the trainer and see that the supply LED glows.

Connect Data output from 4(D1, D2, D3, D4) data outputs to the Data input of the DPSK MdulatorTP7.

2. Connect clock output TP1 to the clock input of the DPSK modulator TP8.
3. Now Connect carrier output TP2 to the carrier input of the DPSK modulator TP10.
4. Observe the Differential Data output on the CRO at TP9 test point as shown on the front panel.
5. Observe the Phase shifted DPSK output waveform on the CRO corresponding to the differential data output.
6. Connect DPSK MODULATOR output TP11 to the DPSK input of the DEMODULATOR TP12.
7. Connect carrier output TP2 to the Carrier input of the DPSK Demodulator TP13.
8. Also connect clock output TP1 to the clock input of the DPSK demodulator TP14.
9. Now observe the DPSK demodulated output waveform TP15 on the CRO.

RESULT: -

Hence DPSK modulation and demodulation is observed and verified its working and also the corresponding output waveforms are plotted.

BLOCK DIAGRAM: -

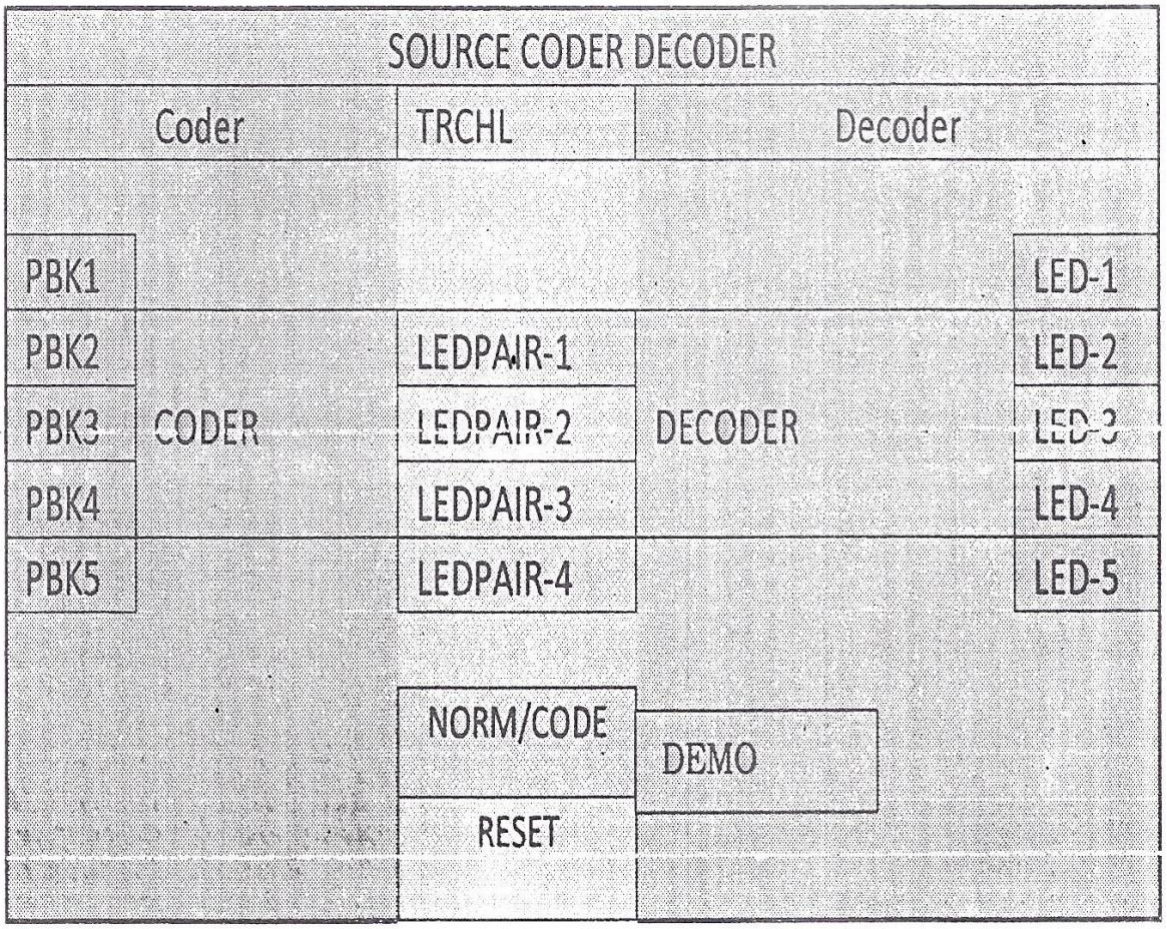


Fig: SOURCE ENCODER AND DECODER

7. SOURCE ENCODER AND DECODER

AIM

To select an information having in-equal probability of occurrence of each symbol or having redundancy in the information and applying a source code using one of the technique i.e Huffman coding, observing the size of the coded information, sending the minimized packet, decoding at the receiving end getting back the full information sent.

EQUIPMENT: -

1. Source Encoder and Decoder Trainer.
2. Oscilloscope 30MHz,Dual Channel.
3. Patch chords.

THEORY

Source Coding is a technique of compressing the source information size based on the probability of occurrence of each information symbol. Decoding is the reverse process to get back the full source information.

In every day we employ this in transferring big files, particularly image/voice files by zipping them and transferring to the destination and unzipping at the destination.

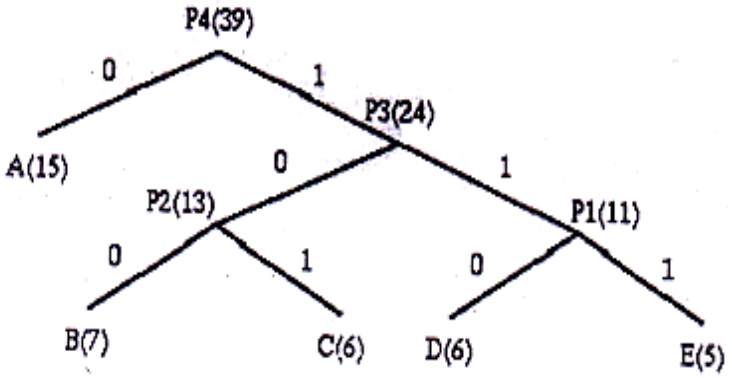
Example: Huffman coding

When we have to transmit a set of symbols over a communication channels, normally choose no. of bits to accommodate all the symbols for example 8 symbols can be coded using 3 bits, because 3 bits give us 3 $2^{\wedge}$ =8 combinations. With this type of normal binary representation we can transfer any combination of symbols in any order. However if our information to be transmitted has a set of symbols but the occurrence of symbols with different frequencies, we can employ a coding technique whereby we choose less no of bits for the frequently occurring symbol and more bits for the less occurring symbol. This way we can represent our information which is a sequence of symbols with less no.of bits i.e. reduced size. In everyday language, we call this zipping.

Huffman coding is based on the frequency of occurrence of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a Code Book which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding.

The Huffman algorithm is now briefly summarized:

A bottom-up approach



Symbol	Count	log(1/p)	Code	Subtotal (# of bits)
A	15	1.38	0	15
B	7	2.48	100	21
C	6	2.70	101	18
D	6	2.70	110	18
E	5	2.96	111	15
			TOTAL (# of bits):	87

1. Initialization: Put all nodes in an OPEN list, keep it sorted at all times (e.g., ABCDE)
2. Repeat until the OPEN list has only one node left:

a) From OPEN pick two nodes having the lowest frequencies/ probabilities, create a parent node of them.

b) Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.

c) Assign code 0,1 to the two branches of the tree, and delete the children from OPEN.

The following points are worth noting the about the above algorithm:

- Decoding for the above two algorithms is trivial as long as the coding table (the statistics) is sent before the data. (There is a bit overhead for sending this, negligible if the data file is big.)
- If prior statistics are available and accurate, then Huffman coding is very good.

In the above example:

Number of bits needed for Huffman Coding is: 87/39=2.23

Examples

Let us take an example that we wish to transmit a word ABRAKADABRA which has 5 types of symbols A, B, R, K, D

In normal coding to represent one of the 5 symbols we need 3 bits.

To transmit this 11 chars we will need $3*11=33$ bits.

However by employing Hoffman source coding as below we can transmit the information using only 23 bits.

B = 100

K = 1010

D = 1011

R = 11

BR K D BRA = 01001101010010110100110

- This is eleven letters in 23 bits.
- A fixed – width encoding would require 3 bits for five different letters, or 33 bits for 11 letters.
- Notice that the encoded bit string can be decoded!

OBSERVATIONS:

Source Symbol	Codeword without coding (Normal mode)	Code word with source coding (Code mode)	Probability of the source symbol
A			
B			
D			
K			
R			
Average codeword length			

Perform Huffman coding here (Manually)

The Decoding Process

The decoding process is just the reverse process (table look up process) i.e. if we receive a pattern 1010 its is decoded as letter C, however one important point is to be noticed that if the input symbols are concatenated i.e put in continuous sequence we should be able to distinguish the no of bits corresponding to each symbol, this is possible only if the code is chosen such that the confusion does not arise, Let us apply the decoding process for the above example sequence received is 010011010100101101001 10

Let us pick the first bit which is 0. Is there a symbol for only one bit as 0? Yes there is a symbol-A for this code. So let us separate this as one symbol so it becomes 0,100. Then further there is no symbol with single bit 1, there is no symbol with 2 bits 10 going further we take 100. There

is a symbol B assigned for 100. So we separate this as 0,100, we continue like this and decode the whole pattern.

If there are errors in the received sequence that we can not decode all the subsequent symbols following the error.

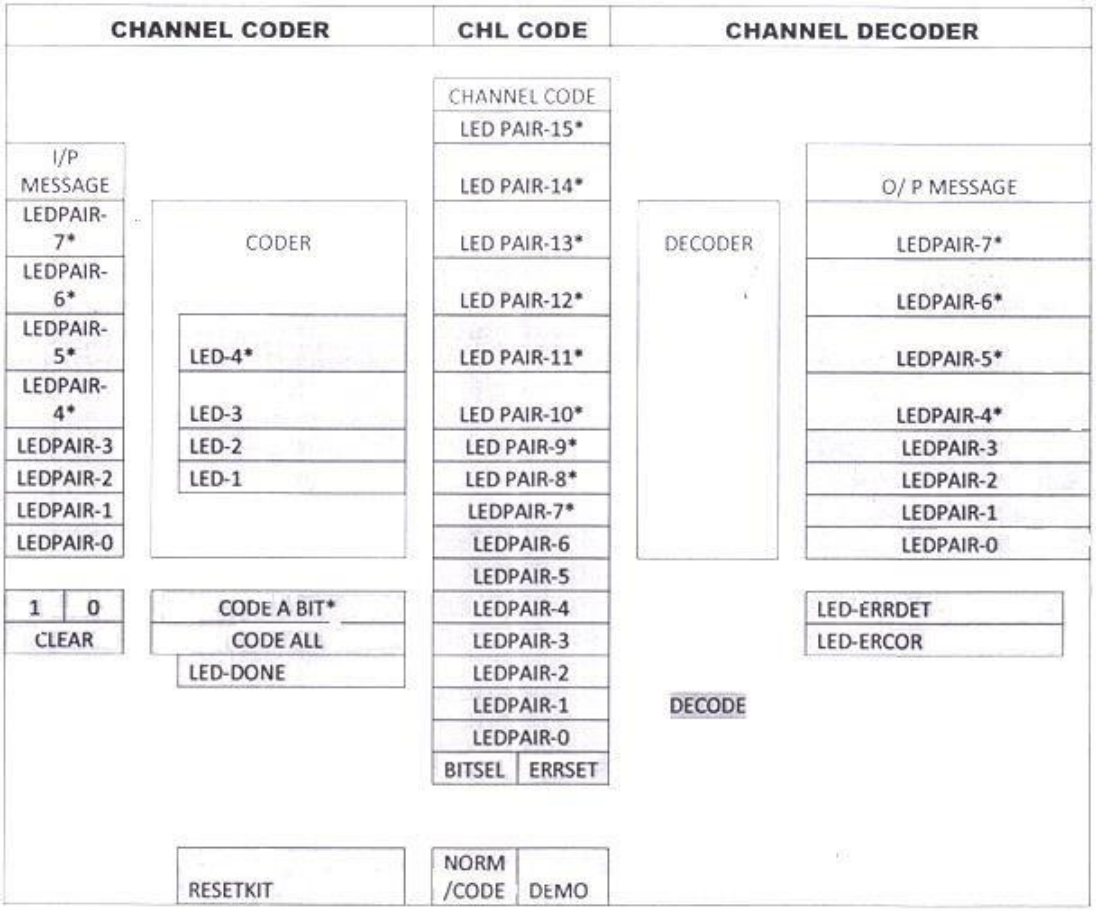
Procedure

1. Observe the signal chain.
2. Then verify how many bits are taken to transmit ABRAKADABRA in normal and source coded mode
3. To send an input symbol, push any one of the input symbol keys, to transmit a letter A, student has to press the key marked A
4. Observer how many bits are being transmitted for this key. And the bit code for the same., and note down the bits being transmitted on the LEDS (1 =Red, 0 =Green)
5. Observe if the corresponding output LED glows corresponding to the symbol pressed at the input.

RESULT:

Hence, Source encoding and decoding techniques are observed.

Block/Circuit Diagram:



8. LINEAR BLOCK CODES ENCODER & DECODER

Aim

To observe that the errors received through a noisy channel can be removed/minimized by employing the error detection and correction code.

Theory

This topic comes under channel coding techniques which are employed for the purpose of detecting and correcting errors occurring in the communication channel. When information is represented in blocks of k bits we can add few extra bits increasing the block size to n bits and employ block coding techniques to detect and correct errors in the reception.

We can add a few extra bits to the information bits i.e. provide some redundancy and detect/correct the errors from the received data. More redundancy we provide more correction we can have. By using hamming coding technique with given no. of extra bits we can extract maximum advantage of detection and correction.

Hamming code (n, k) places symbols represented by n bits having k information bits at a maximum distance from each other allowing us to detect more errors and correct more errors. The decoder if it finds the errors which cannot be corrected, it shows the O/P but indicates that the same with error LED at the output. Normally if in applications using command and control the decoded O/P having errors is not used it is discarded.

Procedure:

1. Generate a 4 bit message in normal mode, encode it and decode it without any error
2. Generate a 4 bit message in normal mode, encode it create a 1 bit or 2 bit error(s) decode the message observe the decoded message at receiver w.r.t. message sent at the transmitter.
3. Generate a 4 bit message in code mode, encode it and decode it without any error, identify the systematic block code word in terms of message and parity bits
4. Repeat the step no. 3 for a single bit error and observe that, the decoder is capable of detecting and correcting 1 bit error
5. Repeat step no. 3 for two bit errors and observe that, the decoder is capable of detecting 2 bit errors and but unable to correct the errors
6. Generate the code book for the all possible 4 bit messages

Observations Table:

	Message	Transmitted code word	Code word with and without errors	Received codeword	Decoded message	Error detecting and correcting capability
Normal mode without error						
Normal mode with 1 bit error						
Normal mode with 2 bit errors						
Code mode without error						
Code mode with 1 bit error						
Code mode with 2 bit erros						

Code Book:

Message	Code Word	Message	Code Word
0 0 0 0		1 0 0 0	
0 0 0 1		1 0 0 1	
0 0 1 0		1 0 1 0	
0 0 1 1		1 0 1 1	
0 1 0 0		1 1 0 0	
0 1 0 1		1 1 0 1	
0 1 1 0		1 1 1 0	
0 1 1 1		1 1 1 1	

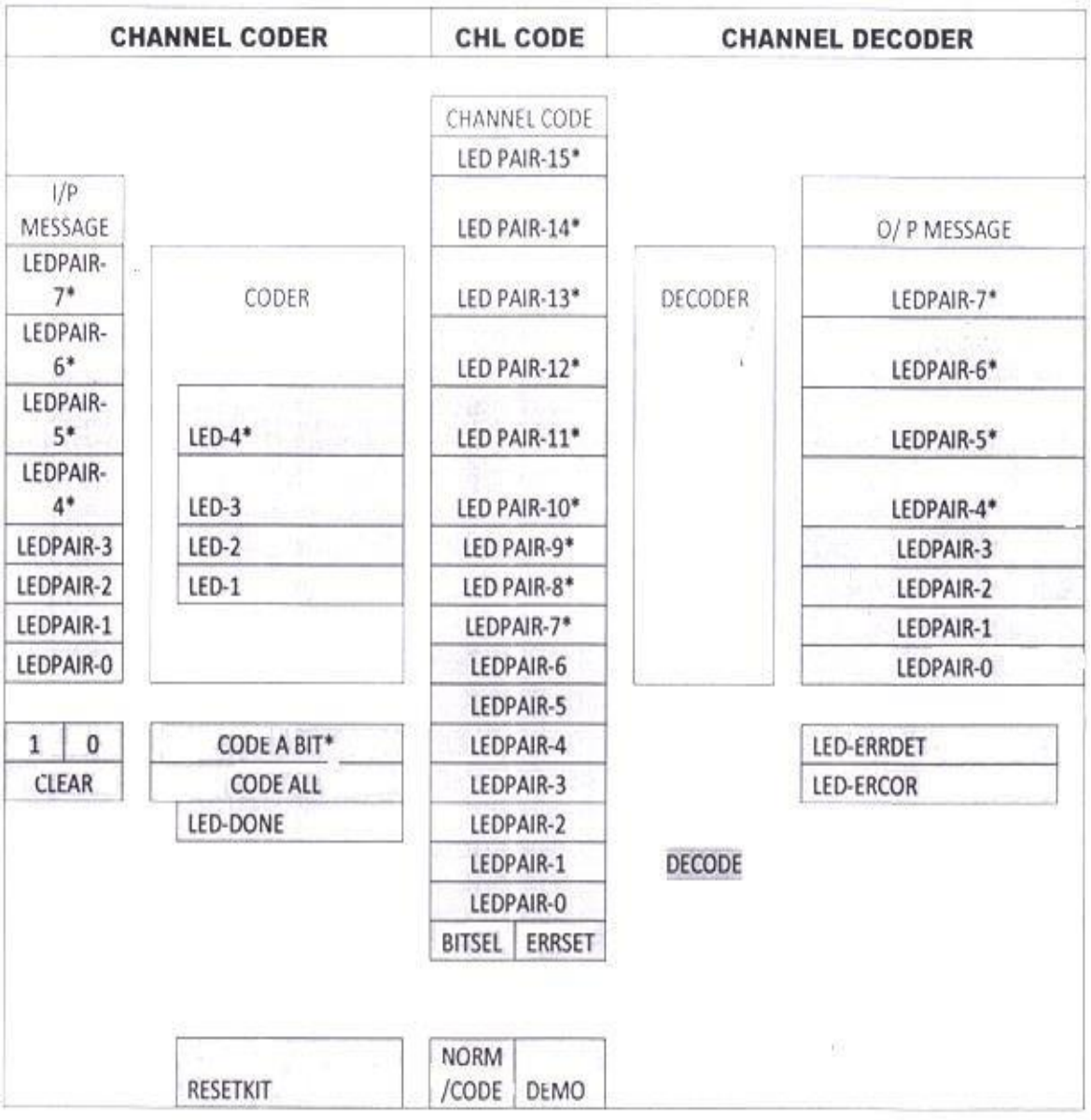
Manual Calculations:

The Generator Matrix: (as per the kit design)

RESULT:

Hence, Linear Block Codes encoding and decoding techniques are observed.

Block/Circuit Diagram:



9. BINARY CYCLIC CODES ENCODER AND DECODER

Aim

To observe that the errors received through a noisy channel can be removed/ minimized by employing the error detection and correction code, by using an algebraic structure.

Theory

In coding theory cyclic codes are the sub-class of linear block error correcting codes that have convenient algebraic structures for efficient error detection and correction. A linear code is called cyclic code if every cycle shift of code vector produces some other code vector i.e. the cycle shift to the data in an array should also represent the data in the same array.
Example: arr {(0000), (0101), (1010), (1111)}

There are two cyclic codes encoding techniques:

- Nonsystematic: Encoded data is obtained by performing M(p) * G(p)/
- Systematic: Coded data ($C_2C_1C_0$) is obtained by performing P^q * M(p)/G(p) and resultant Encoded data will be in form of $M_3M_2M_1M_0C_2C_1C_0$

Here M (p) is input Message / Data of p

G (p) is constant algebraic expression of p

q is constraint length i.e. Systematic (7, 4) gives q=n-k

n is channel length and

k is data length

Let us consider an example for implementation of the experiment (Q) Design the encoder for the Systematic (7, 4) cyclic code generated by a polynomial $G(p)= p^3 + p + 1$ and decode the data transferred by syndrome decoding procedure for knowing input.

The implementation of cyclic encoding and decoding kit is based on the same example. The encoded data output will be $M_3M_2M_1M_0C_2C_1C_0$ which is can be obtained from Systematic coding procedure and the decoding the data will be done by syndrome decoding procedure to regenerate input and correct the data received from any noisy channel.

Observation Table

	Message	Transmitted code word	Code word with and without errors	Received codeword	Decoded message	Error detecting and correcting capability
Normal mode without error						
Normal mode with 1 bit error						
Normal mode with 2 bit errors						
Code mode without error						
Code mode with 1 bit error						
Code mode with 2 bit erros						

Code Book:

Message	Code Word	Message	Code Word
0 0 0 0		1 0 0 0	
0 0 0 1		1 0 0 1	
0 0 1 0		1 0 1 0	
0 0 1 1		1 0 1 1	
0 1 0 0		1 1 0 0	
0 1 0 1		1 1 0 1	
0 1 1 0		1 1 1 0	
0 1 1 1		1 1 1 1	

Procedure:

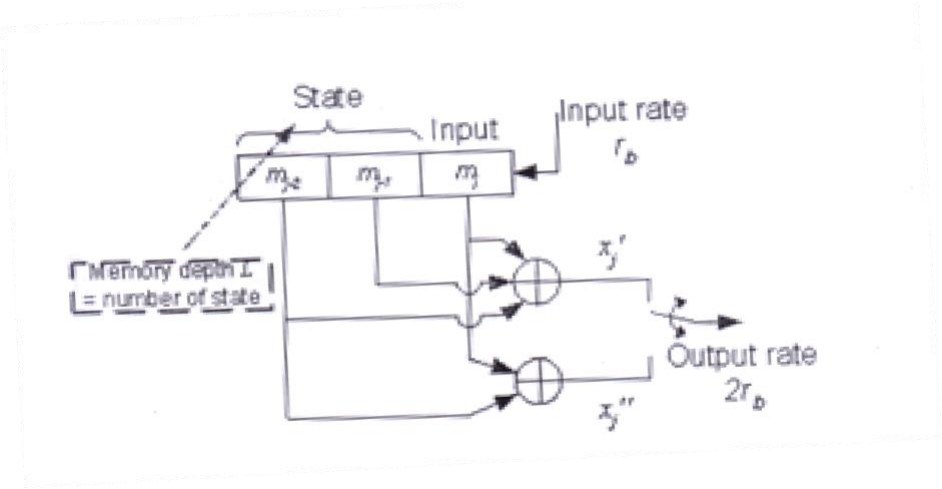
1. Generate a 4 bit message in normal mode, encode it and decode it without any error
2. Generate a 4 bit message in normal mode, encode it create a 1 bit or 2 bit error(s) decode the message observe the decoded message at receiver w.r.t. message sent at the transmitter.
3. Generate a 4 bit message in code mode, encode it and decode it without any error, identify the systematic block code word in terms of message and parity bits
4. Repeat the step no. 3 for a single bit error and observe that, the decoder is capable of detecting and correcting 1 bit error
5. Repeat step no. 3 for two bit errors and observe that, the decoder is capable of detecting 2 bit errors and but unable to correct the errors
6. Generate the code book for the all possible 4 bit messages

Manual Calculations:

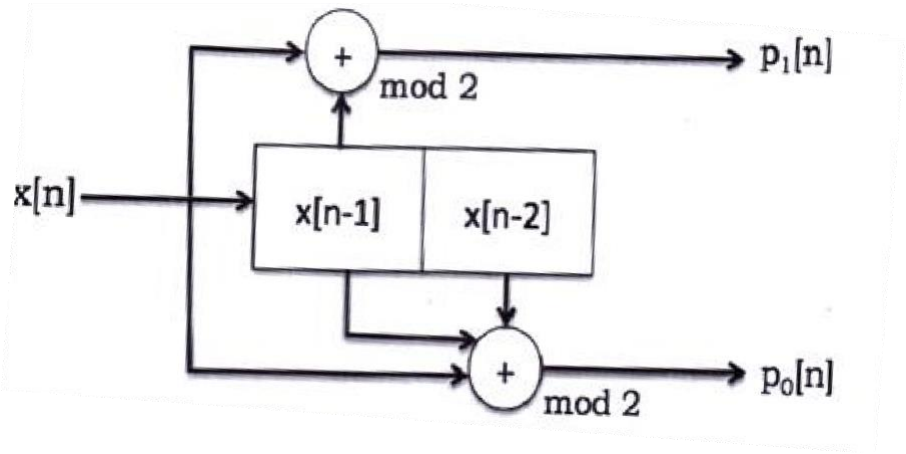
The Generator Polynomial: (as per the kit design)

$$G(x) = 1 + x + x^3$$

Result: Hence, Binary Cyclic Codes encoding and decoding techniques are observed.



(n, k, L)=(2, 1, 2) encoder



$x_j = m_{j-2} \oplus m_{j-1} \oplus m_j$

Implemented Encoder in this Kit

$x_j = m_{j-2} \oplus m_j$

10. CONVOLUTION CODES ENCODER AND DECODER

Aim

Employ one of the convolution codes and observe its error correcting performance and decode-ability.

Theory

Encoder

Convolutional encoder is a finite state machine (FSM), processing information bits in a serial manner. Thus the generated code is a function of input and the states of the FSM. In this (n, k, L)=(2, 1, 2) encoder each message bits influences a span of n(L+1)=6 successive output bits

(a) Rate: Ratio of the number of input bits to the number of output bits. In this example, rate is ½ which means there are two output bits for each input bit.

(b) Constraint length: The number of delay elements in the convolutional coding. In this example, with K = 3 there are two delay elements.

(c) Generator Polynomial: Wiring of the input sequence with the delay elements to form the output. In this example, generator polynomial is $[7, 5]_8$ =[111,101] . The output from the 7_8 =111 arm uses the XOR of the current input, previous input and the previous to previous input. The output from the 5_8 =101 uses the XOR of the current input and the previous to previous input.

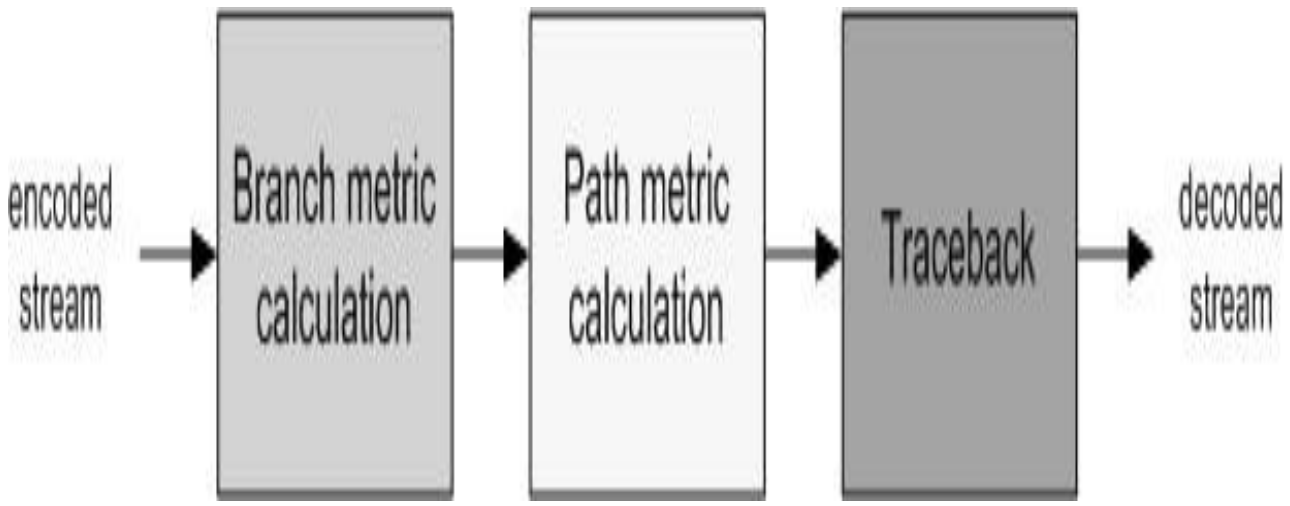
Decoder Using Viterbi Algorithm

Viterbi algorithm reconstructs the maximum-likelihood path given the input sequence.

Let's define some terms:

A soft decision decoder – a decoder receiving bits from the channel with some kind of reliability estimate. Three bits are usually sufficient for this task. Further increasing soft decision width will increase performance only slightly while considerably increasing computational difficulty. For example, if we use a 3-bit soft decision, then “000” is the strongest zero, “011” is a weakest zero, “100” is a weakest one and “111” is a strongest one.

A hard decision decoder – a decoder which receives only bits from the channel (without any reliability estimate).



Viterbi decoder data flow.

A branch metric – a distance between the received pair of bits and one of the "ideal" pairs ("00", "01", "10", "11").

A path metric – a sum of metrics of all branches in the path.

A meaning of distance in this context depends on the type of the decoder:

for a hard decision decoder it is a Hamming distance, i.e. a number of differing bits;

for a soft decision decoder it is an Euclidean distance.

In these terms, the maximum-likelihood path is a path with the minimal path metric. Thus the problem of decoding is equivalent to the problem of finding such a path.

Let's suppose that for every possible encoder state we know a path with minimum metric ending in this state. For any given encoder state there is two (and only two) states from which the encoder can move to that state, and for both of these transitions we know branch metrics. So, there are only two paths ending in any given state on the next step. One of them has lesser metric, it is a survivor path. The other path is dropped as less likely. Thus we know a path with minimum metric on the next step, and the above procedure can be repeated.

Implementation

A Viterbi algorithm consists of the following three major parts:

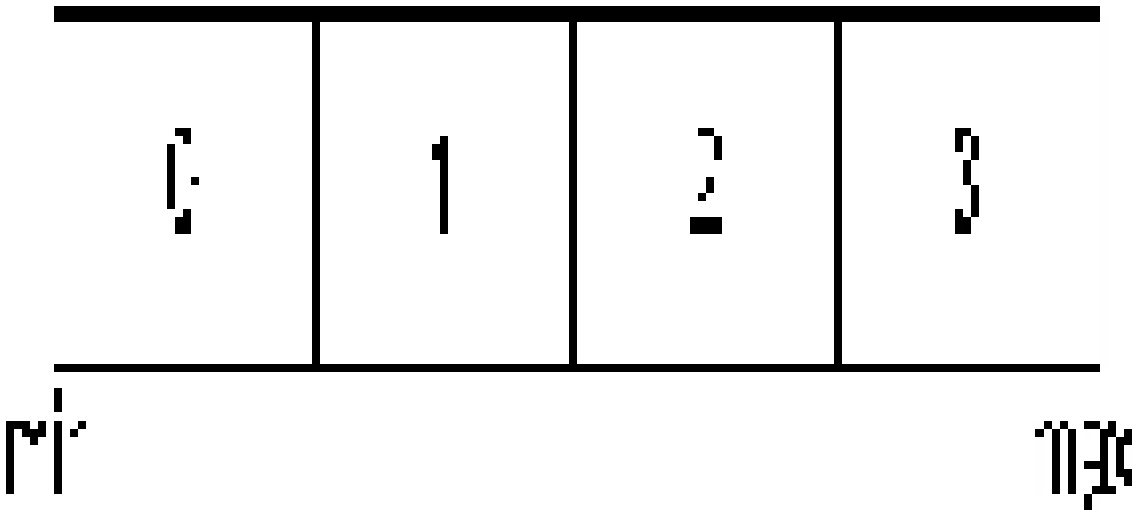
1. Branch metric calculation – calculation of a distance between the input pair of bits and the four possible "ideal" pairs ("00", "01", "10", "11").
2. Path metric calculation – for every encoder state, calculate a metric for the survivor path ending in this state (a survivor path is a path with the minimum metric).
3. Traceback – this step is necessary for hardware implementations that don't store full information about the survivor paths, but store only one bit decision every time when one survivor path is selected from the two.

Branch Metric Calculation

Methods of branch metric calculation are different for hard decision and soft decision decoders.

For a hard decision decoder, a branch metric is a Hamming distance between the received pair of bits and the "ideal" pair. Therefore, a branch metric can take values of 0, 1 and 2. Thus for every input pair we have 4 branch metrics (one for each pair of "ideal" values).

For a soft decision decoder, a branch metric is measured using the Euclidean distance. Let x be the first received bit in the pair, y – the second, x_0 and y_0 – the "ideal" values. Then branch metric is



A modulo-normalization approach for path metrics

$M_b=(x-x_0)^2 + (y-y_0)^2$.

Furthermore, when we calculate 4 branch metric for a soft decision decoder, we don't actually need to know absolute metric values – only the difference between them makes sense. So, nothing will change if we subtract one value from the all four branch metrics:

$M_b=(x^2-2xx_0+x_0^2)+(y^2-2yy_0+y_0^2)$;

$M_b^* = M_b - x^2 - y^2=(x^2-2xx_0)+(y^2-2yy_0)$.

Note that the second formula, M_b^*, can be calculated without hardware multiplication: x_0^2 and y_0^2 can be pre-calculated, and multiplication of x by x_0 and y by y_0 can be done very easily in hardware given that x_0 and y_0 are constants.

It should be also noted that M_b^* is a signed variable and should be calculated in 2's complement format.

Path Metric Calculation

Path metrics are calculated using a procedure called ACS (Add-Compare-Select). This procedure is repeated for every encoder state.

1. Add – for a given state, we know two states on the previous step which can move to this state, and the output bit pairs that correspond to these transitions. To calculate new path metrics, we add the previous path metrics with the corresponding branch metrics.
2. Compare, select – we now have two paths, ending in a given state. One of them (with the greater metric) is dropped.

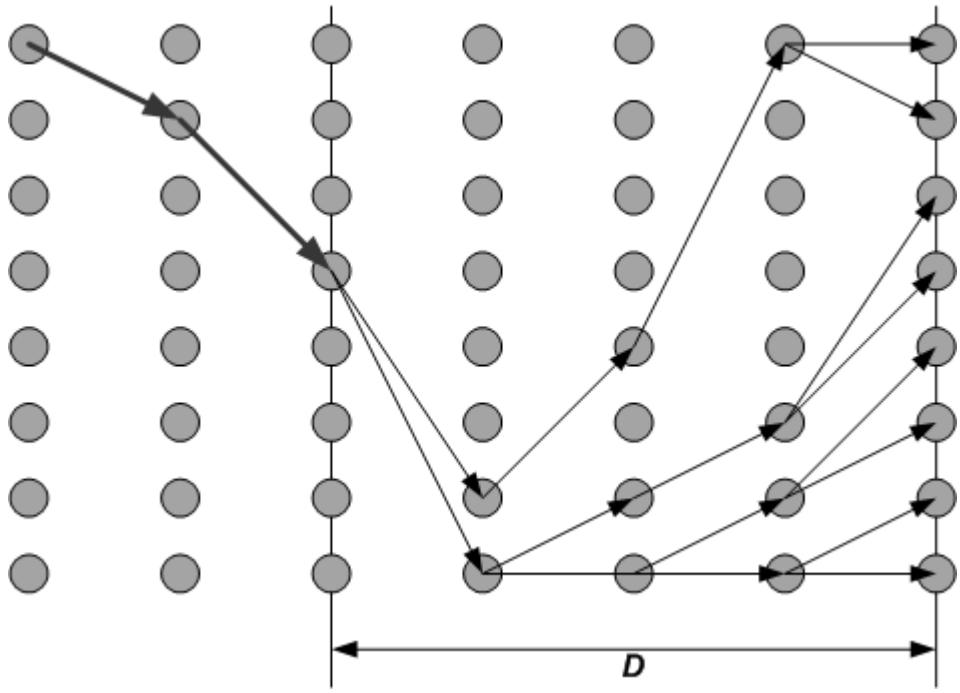
As there are 2^{K-1} encoder states, we have 2^{K-1} survivor paths at any given time.

It is important that the difference between two survivor path metrics cannot exceed $\delta\log(K-1)$, where δ is a difference between maximum and minimum possible branch metrics.

The problem with path metrics is that they tend to grow constantly and will eventually overflow. But, since the absolute values of path metric don't actually matter, and the difference between them is limited, a data type with a certain number of bits will be sufficient.

There are two ways of dealing with this problem:

1. Since the absolute values of path metric don't actually matter, we can at any time subtract an identical value from the metric of every path. It is usually done when all path metrics exceed a chosen threshold (in this case the threshold value is subtracted from every path metric). This method is simple, but not very efficient when implemented in hardware.
2. The second approach allows overflow, but uses a sufficient number of bits to be able to detect whether the overflow took place or not. The compare procedure must be modified in this case.



Survivor paths graph example. Blue circles denote encoder states. It can be seen that all survivor paths have a common beginning (red) and differ only in their endings.

OBSERVATION TABLE

NORMAL SYSTEM WITHOUT CODING				
I/P Data Set	Normal Transmission		Errors in Transmission	
	Chal data	O/P Data	Chal data	O/P Data
1010	1010	1010	1000	1000
SYSTEM WITH CONVOLUTION CODING				
	Changed i/p	Chal data	Chal Data	O/P Data
1010	101000*	111101111000**	111100110000	101000

*Channel input increases such that the memory states should become Zero's** Channel data as per the Convolution encodi

The whole range of the data type's capacity is divided into 4 equal parts. If one path metric is in the 3-rd quarter, and the other – in the 0-th, then the overflow took place and the path in the 3-rd quarter should be selected. In other cases an ordinary compare procedure is applied. This

works, because a difference between path metrics can't exceed a threshold value, and the range of path variable is selected such that it is at least two times greater than the threshold.

Traceback

It has been proven that all survivor paths merge after decoding a sufficiently large block of data (D on Figure 5), i.e. they differ only in their endings and have the common beginning.

If we decode a continuous stream of data, we want our decoder to have finite latency. It is obvious that when some part of path at the beginning of the graph belongs to every survivor path, the decoded bits corresponding to this part can be sent to the output. Given the above statement, we can perform the decoding as follows:

1. Find the survivor paths for N+D input pairs of bits.
2. Trace back from the end of any survivor paths to the beginning.
3. Send N bits to the output.
4. Find the survivor paths for another N pairs of input bits.
5. Go to step 2.

In these procedure D is an important parameter called decoding depth. A decoding depth should be considerably large for quality decoding, no less then 5K. Increasing D decreases the probability of a decoding error, but also increases latency.

As for N, it specifies how many bits we are sending to the output after each trace back. For example, if N=1, the latency is minimal, but the decoder needs to trace the whole tree every step. It is computationally ineffective. In hardware implementations N usually equals D.

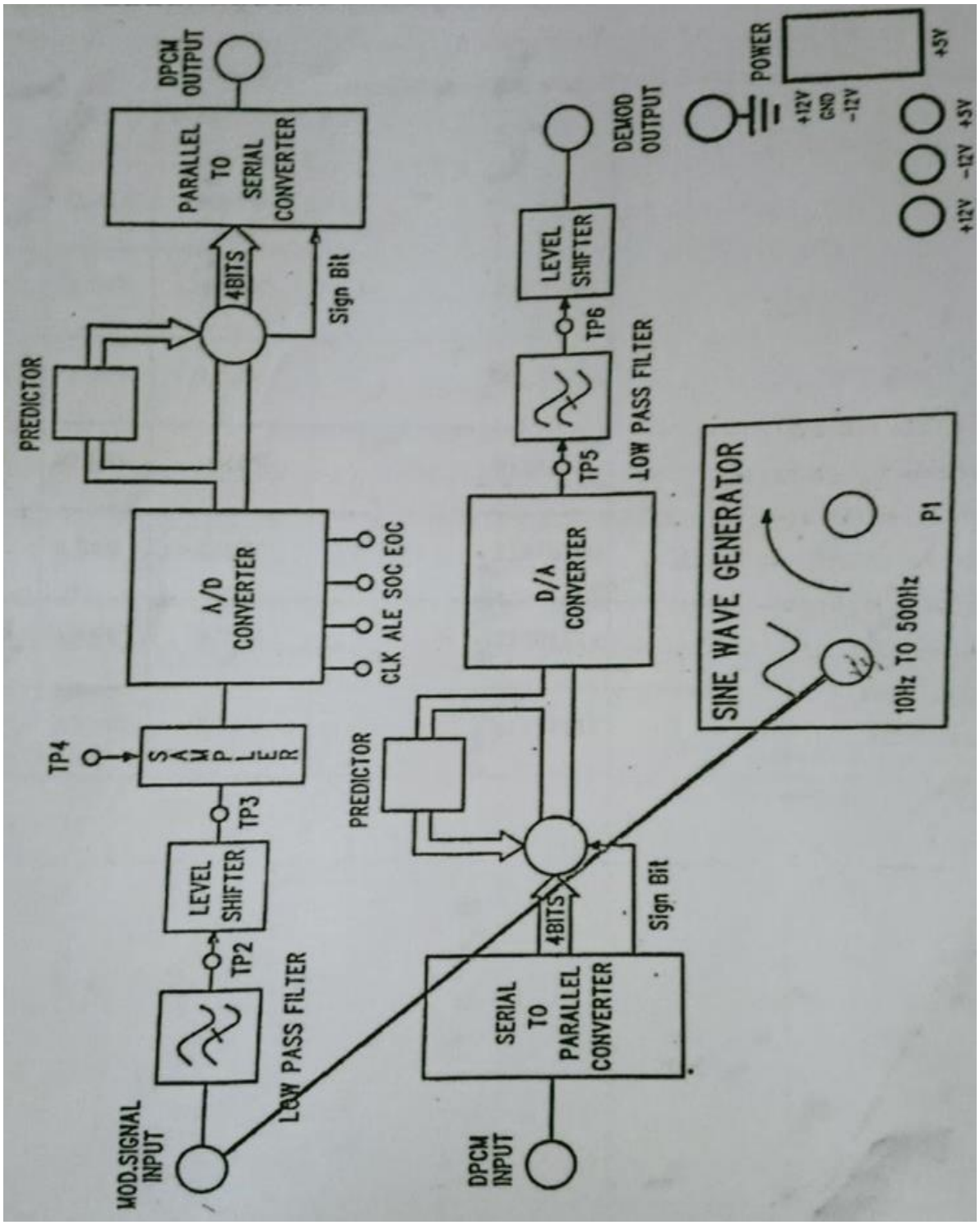
Procedure:

1. Observe the signal chain, i.e. the input stage, coding stage, transmission stage and the decode stage
2. Put the mode selection switch in NORMAL mode and see the process and observe output
3. Student selects input message that is to be coded, by shifting the bits 0/1 by means of pressing the keys 0, 1, CLEAR
4. Student codes this input message by pushing the key 'CODE A BIT' or by pushing 'CODEALL'
5. Now the message is coded and displayed in the transmission path. Student can now introduce an error in the transmission channel by means of pressing the keys BITSEL and ERRSET. On every push of the BITSEL one bit is selected in the channel code, the selected bit will be completely in OFF mode at this stage, if the student presses ERRSET key, the OFF mode bit will be inverted to make it as an error.
6. Now the student pushes the DECODE key, the channel code is decoded and displayed as the Output message. If an error is detected in the channel code ERRDETECTED LED glows,
7. if an error is corrected from channel code then the ERRCORRECTED LED glows in the decoder Output stage.
8. Now put mode selection in CODE mode and repeat the process and observe the output changes.
9. Observe how the error detection and correction in code mode and hence the implementation of convolution encoder and decoder.
10. Pushing Demo mode switch will give a brief description of implementation of KIT

RESULT: Hence, Convolution Codes Encoding And Decoding techniques are observed.

AUGMENTED EXPERIMENTS

BLOCK DIAGRAM:



.

DIFFERENTIAL PULSE CODE MODULATION & DEMODULATION

1. DIFFERENTIAL PULSE CODE MODULATION & DEMODULATION.

Aim:

To Study Differential Pulse Code Modulation And Demodulation By Sending Fixed 1khz Sine And Variable Frequency Sine And Square Waves

Apparatus:

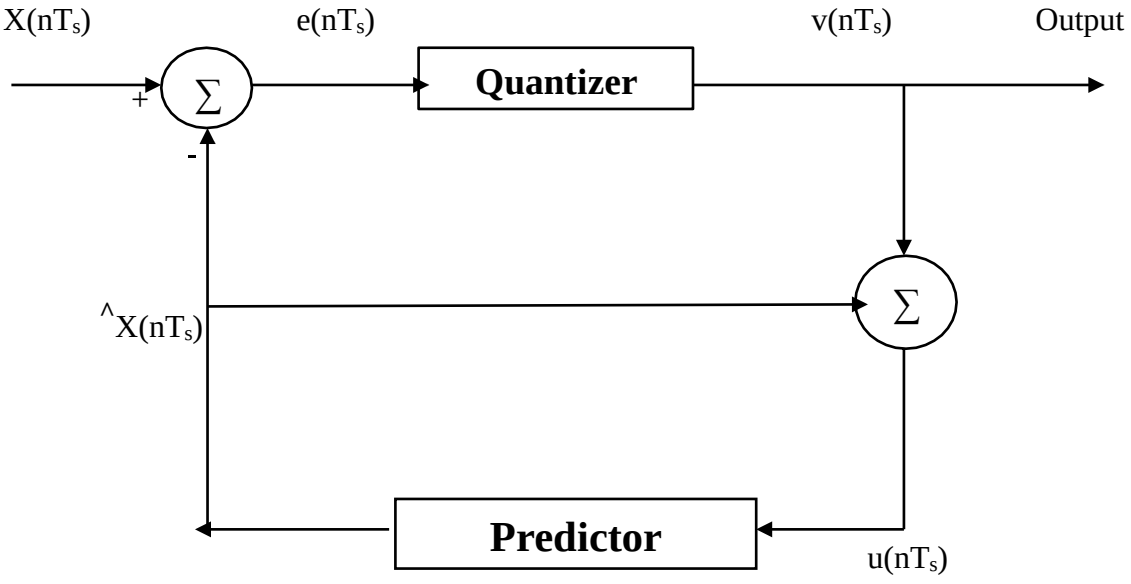
1. DPCM Modulation & Demodulation Trainer Kit.
2. 30 MHz Dual Channel CRO.

Theory:

For the signal which does not change rapidly from one sample to next sample, the PCM scheme is not preferred. When such highly correlated samples are encoded the resulting encoded signal contains redundant information. By removing this redundancy before encoding an efficient coded signal can be obtained. One of such scheme is the DPCM technique. By knowing the past behavior of a signal up to a certain point in time, it is possible to make some inference about the future values. The transmitter and receiver of the DPCM scheme is shown in the below figures. If we know the past behavior of a signal up to a certain point in time, we may use prediction to make an estimate of a future value of the signal. Suppose, a base band signal m (t) is sampled at the rate $f_s = 1/T_s$ to produce the sequence {m (n)} whose samples are T_s seconds apart. It is possible to predict future values of the signal m (t), provides motivation for the differential quantization scheme shown in fig. below.

Transmitter:

Sampled Input



Block diagram of DPCM Trasmitter

OBSERVATIONS:

S.no		Amplitude	Time Period	Frequency
1	Modulating Signal I/P			
2	Actual I/P to the A/D Converter			
3	Sampling clock			
4	DPCM Output			
5	Demodulation O/P to the D/A Converter			
6	LPF Output			

Let x (1) be the signal to be sampled and x(nT's) be it's samples. In this scheme the input to the quantizer is a signal

$$e\ (nts) = x(nT's) - (nT's)$$

Where a(nTs) is the prediction for un quantized sample x(nts). This predicted value is produced by using a predictor whose input, consists of a quantized versions of the input signal x(nt's). The signal e(nT's) is called the prediction error.

By encoding the quantizer output, in this method, we obtain a modified version of the PCM called differential pulse code modulation (DPCM).

Quantizer output,

$$v(nts) = Q[e(nT's)] = e(nt's) + q\ (nT's)$$

Where q (nT's) is the quantization error.

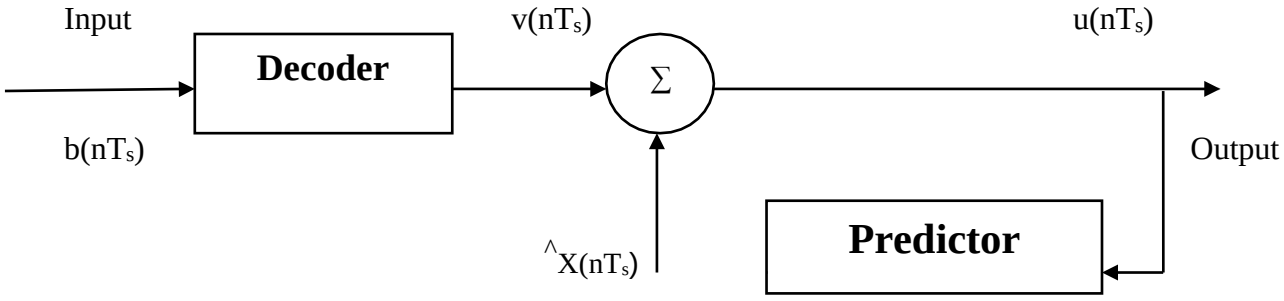
Predictor input is the sum of quantizer output and predictor output,

$$u\ (nt's) = *(nT's) + v(nT's)$$

Using the above equations

$$u\ (nt's) = *(nt's) + e(nt's) + 9(nt's)$$

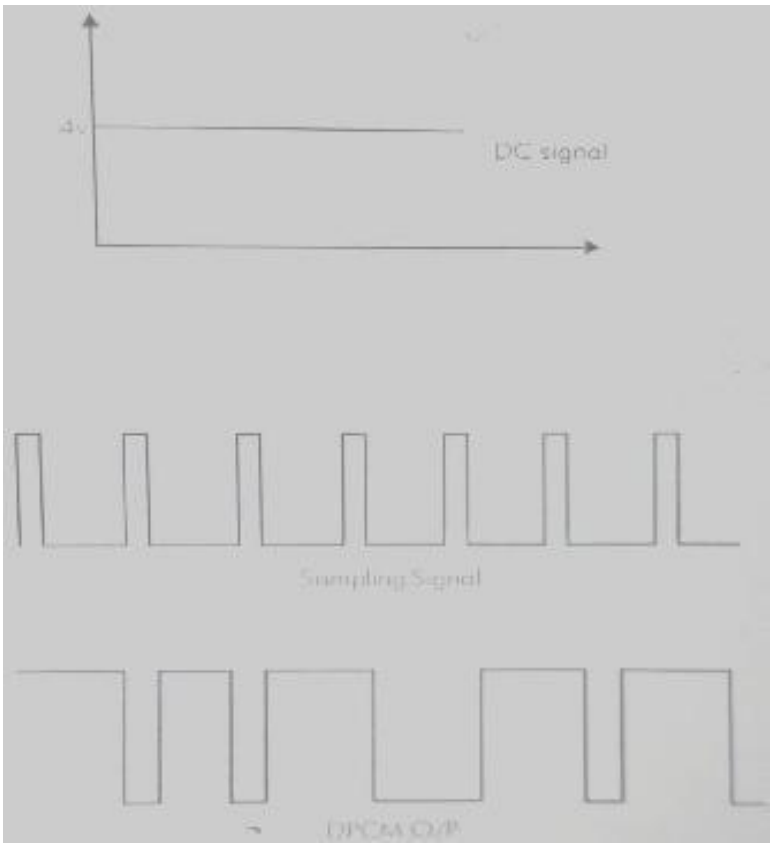
$$u\ (nts) = x(nT's) + q(nT's)$$

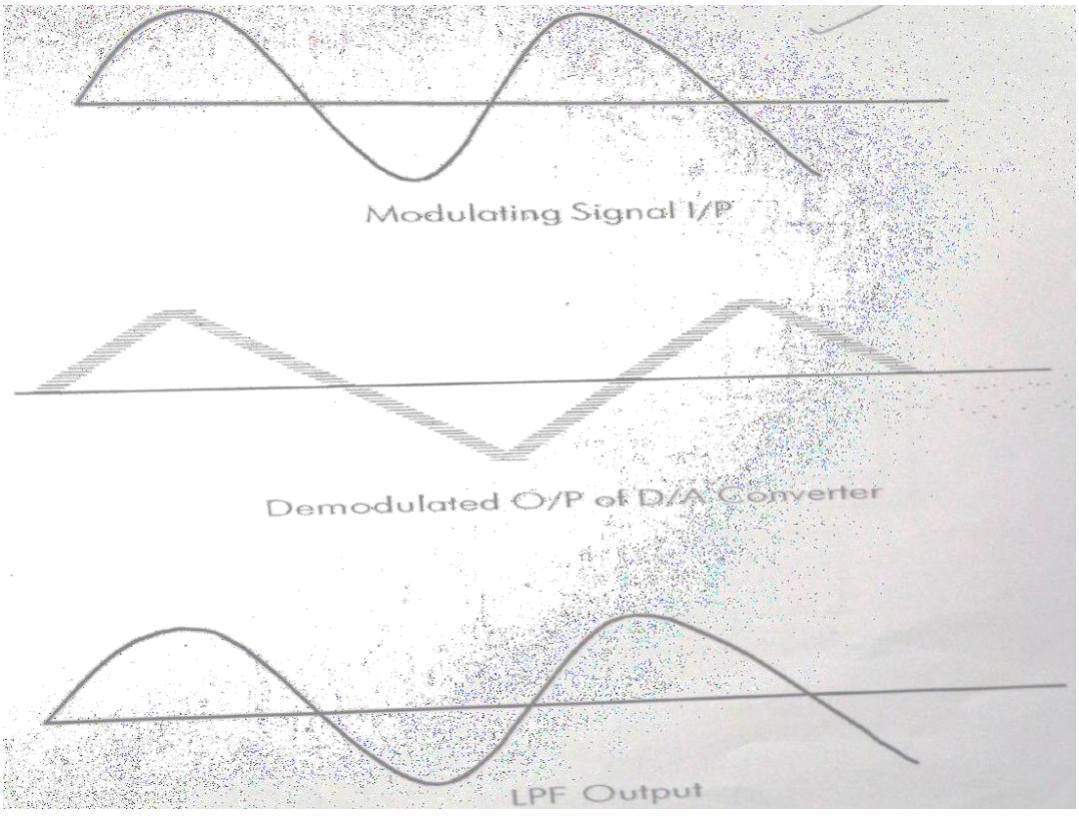


Block diagram of DPCM Receiver.

The receiver consists of a decoder to reconstruct the quantized error signal. The quantized version of the original input is reconstructed from the decoder output using the same predictor as used in the transmitter. In the absence of noise the encoded signal at the receiver input is identical to the encoded signal at the transmitter output. Correspondingly the receive output is equal to u(nts), which differs from the input x(nT's) only by the quantizing error q(nT's).

Model Wave Forms:





Procedure:

1. Connect the CRO probe at Mod signal input and Observe 1KHz sine wave in CRO
2. Connect the probe at TP4 and observe the 4 KHz clock.. This is the sampling rate for the system then change probe to CLK, Observe 2MHz clock this is the system clock for ADC.
3. Connect the patch cord between SINEWAVE and MOD. Input (m (0)).Connect the patch cord between DPCM output and DEMOD Input.
4. Connect CRO channel 1 to sine wave and channel 2 to demod output. Now we can observe modulating input and demodulated output in CRO.
5. Connect Channel 1 at sine wave o/p and channel 2 at demod output, observe waveforms.

Vary the input frequency from 20Hz to 1 KHz, Observe output at demod output. When input is at low frequency, then output is having high amplitude than when input is at high frequency.

Result: Hence The Differential Pulse Code Modulation & Demodulation Techniques Are Observed.

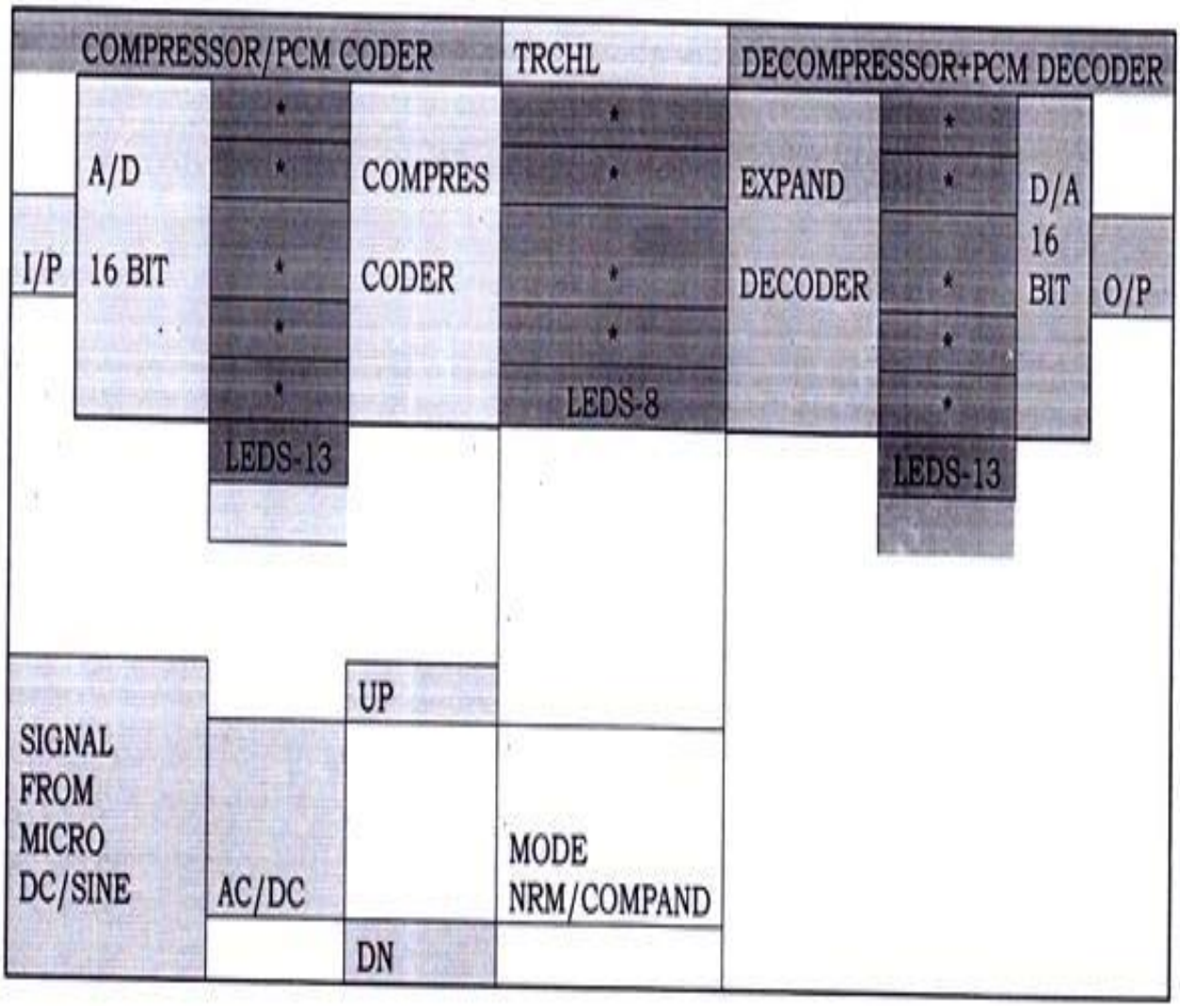


Fig: COMPANDING

2. COMPANDING

Aim

To note down the dynamic range of the signal and S/Q without companding and with companding (both A-law and the u-law) and to observe the improvement in dynamic range and the noise performance with companding.

APPARATUS: 1. Companding Trainer. 2. Oscilloscope – 30 MHz Dual Channels. 3. Patch cards.**Theory**

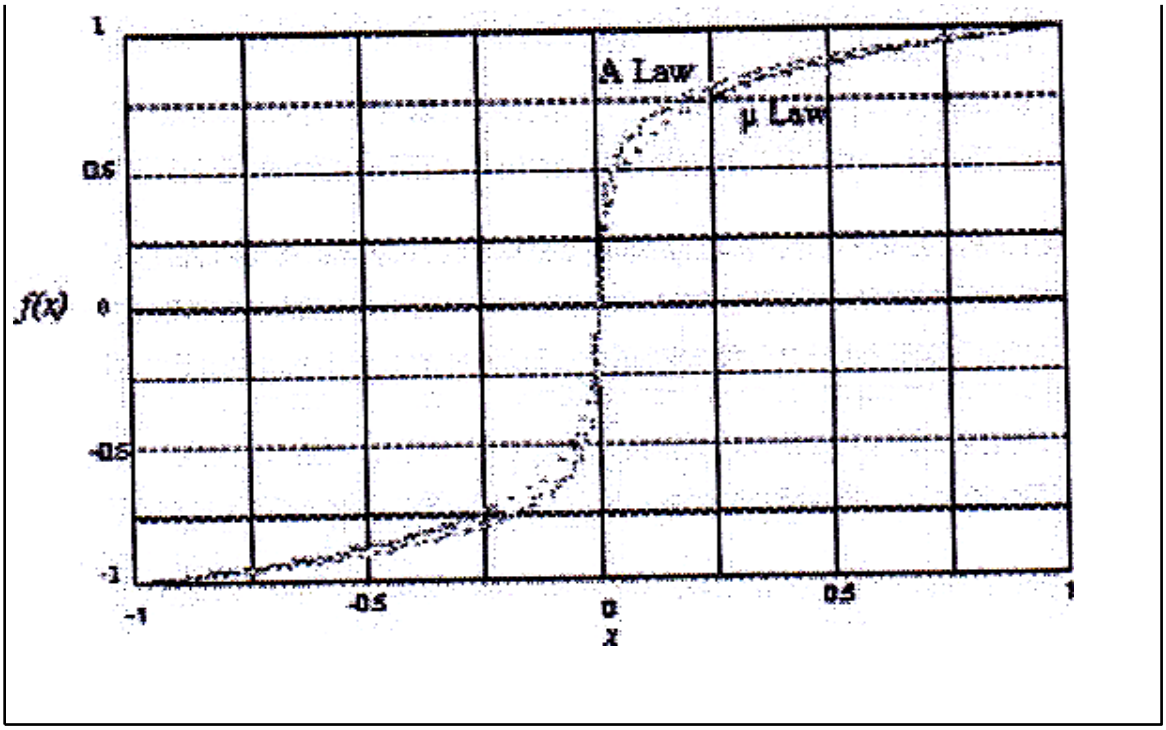
While coding the signal we give higher resolution at low levels and and lower resolution at higher levels. This makes the signal compressed in the code domain.

By applying the reverse mapping at the decoding end we get back the original signal.

By allocating more bits at lower signal levels we can reduce the quantization noise (the noise created by the LSB).

The μ-law and A-law algorithms encode 14-bit and 13-bit signed linear PCM samples (respectively) to logarithmic 8-bit samples. Thus, the G.711 encoder will create a 64 kbit/s bit stream for a signal sampled at 8 kHz.[1]

G.711 μ-law tends to give more resolution to higher range signals while G.711 A-law provides more quantization levels at lower signal levels.



A-Law Binary Encoding Table

Input Values												Compressed Code Word						
												Chord			Step			
bit: 11	10	9	8	7	6	5	4	3	2	1	0	bit: 6	5	4	3	2	1	0
0	0	0	0	0	0	0	a	b	c	d	x	0	0	0	a	b	c	d
0	0	0	0	0	0	1	a	b	c	d	x	0	0	1	a	b	c	d
0	0	0	0	0	1	a	b	c	d	x	x	0	1	0	a	b	c	d
0	0	0	0	1	a	b	c	d	x	x	x	0	1	1	a	b	c	d
0	0	0	1	a	b	c	d	x	x	x	x	1	0	0	a	b	c	d
0	0	1	a	b	c	d	x	x	x	x	x	1	0	1	a	b	c	d
0	1	a	b	c	d	x	x	x	x	x	x	1	1	0	a	b	c	d
1	a	b	c	d	x	x	x	x	x	x	x	1	1	1	a	b	c	d

A-Law Binary Decoding Table

Compressed Code Word							Biased Output Values										
Chord			Step														
bit: 6	5	4	3	2	1	0	bit: 11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	a	b	c	d	0	0	0	0	0	0	0	a	b	c	d	1
0	0	1	a	b	c	d	0	0	0	0	0	0	1	a	b	c	d	1
0	1	0	a	b	c	d	0	0	0	0	0	1	a	b	c	d	1	0
0	1	1	a	b	c	d	0	0	0	0	1	a	b	c	d	1	0	0
1	0	0	a	b	c	d	0	0	0	1	a	b	c	d	1	0	0	0
1	0	1	a	b	c	d	0	0	1	a	b	c	d	1	0	0	0	0
1	1	0	a	b	c	d	0	1	a	b	c	d	1	0	0	0	0	0
1	1	1	a	b	c	d	1	a	b	c	d	1	0	0	0	0	0	0

μ-Law Binary Encoding Table

Biased Input Values													Compressed Code Word						
													Chord			Step			
bit: 12	11	10	9	8	7	6	5	4	3	2	1	0	bit: 6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	a	b	c	d	x	0	0	0	a	b	c	d
0	0	0	0	0	0	1	a	b	c	d	x	x	0	0	1	a	b	c	d
0	0	0	0	0	1	a	b	c	d	x	x	x	0	1	0	a	b	c	d
0	0	0	0	1	a	b	c	d	x	x	x	x	0	1	1	a	b	c	d
0	0	0	1	a	b	c	d	x	x	x	x	x	1	0	0	a	b	c	d
0	0	1	a	b	c	d	x	x	x	x	x	x	1	0	1	a	b	c	d
0	1	a	b	c	d	x	x	x	x	x	x	x	1	1	0	a	b	c	d
1	a	b	c	d	x	x	x	x	x	x	x	x	1	1	1	a	b	c	d

μ-Law Binary Decoding Table

Compressed Code Word							Biased Output Values												
Chord			Step																
bit: 6	5	4	3	2	1	0	bit: 12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	a	b	c	d	0	0	0	0	0	0	0	1	a	b	c	d	1
0	0	1	a	b	c	d	0	0	0	0	0	0	1	a	b	c	d	1	0
0	1	0	a	b	c	d	0	0	0	0	0	1	a	b	c	d	1	0	0
0	1	1	a	b	c	d	0	0	0	0	1	a	b	c	d	1	0	0	0
1	0	0	a	b	c	d	0	0	0	1	a	b	c	d	1	0	0	0	0
1	0	1	a	b	c	d	0	0	1	a	b	c	d	1	0	0	0	0	0
1	1	0	a	b	c	d	0	1	a	b	c	d	1	0	0	0	0	0	0
1	1	1	a	b	c	d	1	a	b	c	d	1	0	0	0	0	0	0	0

The Basic Building Blocks of The Companding

Signal Generator:

The kit generates -3 to +3 DC Voltage at the input by using the UP/DOWN keys in DC mode.

The kit also generates a fixed ac waveform when the switch is in AC Mode.

A/D Convertor:

This is a 16 bit A/D convertor, for Companding we need the following sizes of A/D, for linear 8 bit, for A law 13 bit, for u law 14 bit.

The most significant bit is used for signal sign, bit is 1 means negative.

From the 16 bit convertor we get the sign bit and 15 bit magnitude

For linear case we take the most significant 7 bits from the magnitude and the sign bit to sign bit.

For A law we take sign bit sign bit, most significant 12 bit as magnitude of the input.

For u law we are showing only the magnitude bits on LEDS. The sign bit is not shown.

The compressor:

The compressor converts the 12 bit magnitude to coded 7 bits (A law). These 7 bits along with the sign bit go out on the channel of communication.

The Expander:

The expander converts the 7 coded magnitude back to the original 12 bit magnitude (A law).

D/A:

The D/A is 16 bit implementation, the expander output sign bit gets loaded as sign bit, the magnitude 12 bits get loaded to the most significant 12 bits of the D/A.

The Key switches and controls on the Panel

UP/DOWN KEYS: In dc experiment mode, when up key is pressed once the dc level goes up by one step as per the experiment table. Similarly one press of DOWN key reduces the dc level by one step. When you want bring the dc level back to start value of 0 Press Reset Key once.

Reset Key: Puts the kit in the dc experiment mode with the starting dc level set at 0

AC/DC Slide Switch: This switch puts the kit either in DC mode or AC mode. In DC mode user can give apply a given DC with up/down keys. Whereas in AC mode the kit generates a sample AC wave form at the input and passes the same through the signal chain, user has to observe the AC waveform at the output in normal/companded mode.

Normal/Compand Switch: This slide switch puts the signal chain for normal 8 bit transmit channel or companded 8 bit chl. In normal mode the upper 8 bits of the A/D and D/A are used, the lower bits are ignored. In companded mode as per the Companding law A/u law the 13/14 bits of A/D and D/A are used.

For the same input the user can change the signal chain from Normal to Compand and observe the difference in the output.

Demo Key:

When this key is pressed the kit goes into dc demo mode. In this mode input changes in steps automatically and the corresponding A/D, Channel, D/A Leds are shown and the analogue output is obtained. To come out of this mode press the Reset key once.

SelfTest Key:

When this key is pressed the kit goes into Self Test mode. In this mode ALL LEDs are tested, keys are tested and analogue chain is calibrated. This mode is normally used in the factory for testing and calibration. However if the user doubts the LEDS he can use this mode check out if all LEDS are able to glow or not. To come out of this mode press reset key once.

A-law/u-Law Switch:

This switch come into effect when companding is ON, once companding is ON this switch selects the companding law to be used.).

Observations Table

For u law

I/P DC Voltage	A/D O/P 12 bit	Ch1 Code W/O Compdr	Ch1 Code With Compdr u-law	D/A I/P 12 bit	O/P Voltage	Error Digital	Q/S
A	B	C	D	E	F	E-B	(E-B)/B

For A-law:

I/P DC Voltage	A/D O/P 12 bit	Ch1 Code W/O Compdr	Ch1 Code With Compdr A-law	D/A I/P 12 bit	O/P Voltage	Error Digital	Q/S
A	B	C	D	E	F	E-B	(E-B)/B
0.000	0×000	0×00	0×00				
0.012	0×010	0×00	0×10				
0.023	0×01F	0×00	0×1F				
0.034	0×02E	0×01	0×17				
0.045	0×03E	0×01	0×1F				
0.070	0×060	0×03	0×28				
0.091	0×07C	0×03	0×2F				
0.750	0×400	0×20	0×C0				
1.453	0×7C0	0×3E	0×6F				
1.500	0×800	0×40	0×70				
2.250	0×C00	0×60	0×78				
2.906	0×F80	0×7C	0×7F				

Procedure:

1. First observe the communication blocks in the signal chain
2. Apply a given dc voltage at the input by using the up/down keys, measure this with multi meter.
3. Note down the codes and the voltages as per the table given below.
4. Do this for both the linear mode and companded mode (A Law
5. Observe that higher Quantization error Q/S in the case of linear mode compared to the companded mode.
6. Observe the quality improvement of a companded channel for a low level ac signal given by the kit itself in AC mode.
7. Observe the improvement in wave shape for a low level ac waveform by putting the kit in AC mode.
8. Compare the waveforms.
9. Connect I/P wave form to DSO channel-1 (trigger source ch 1)
10. Connect O/P waveform to DSO channel-2
11. Observe the variation in channel-2, by putting the mode switch in companding and normal 8 bit linear chl mode.

Result:

Hence, observed the companding technique and the values are noted with companding and without companding to different DC voltages.

FREQUENCY SHIFT KEYING MODULATION & DEMODULATION USING MATLAB

AIM:

To write a program to perform FSK using MATLAB and to obtain its output waveforms.

APPARATUS REQUIRED:

Personal computer, MATLAB R2016a

PROCEDURE:

1. Click start, select All programs, select MATLAB R2016a
2. Click file, select new M-file
3. Type the program
4. Save it with extension.m
5. Click RUN to obtain the output waveform

PROGRAM:

FSK

```
clc %for clearing the command window
close all %for closing all the window except command window clear all %for deleting all the variables from the memory
fc1=input('Enter the freq of 1st Sine Wave carrier:'); fc2=input('Enter the freq of 2nd Sine Wave carrier:');
 fp=input('Enter the freq of Periodic Binary pulse (Message):');
amp=input('Enter the amplitude (For Both Carrier & Binary Pulse Message):');
amp=amp/2;
t=0:0.001:1;
c1=amp.*sin(2*pi*fc1*t);
c2=amp.*sin(2*pi*fc2*t);
subplot(5,1,1);
plot(t,c1)
xlabel('Time')
ylabel('Amplitude')
title('Carrier 1 Wave')
subplot(5,1,2) plot(t,c2)
xlabel('Time')
ylabel('Amplitude')
title('Carrier 2 Wave')
m=amp.*square(2*pi*fp*t)+amp;
subplot(5,1,3)
plot(t,m) xlabel('Time')
ylabel('Amplitude')
title('Binary Message Pulses') for i=0:1000
if m(i+1)==0 mm(i+1)=c2(i+1);
```

WAVEFORM:

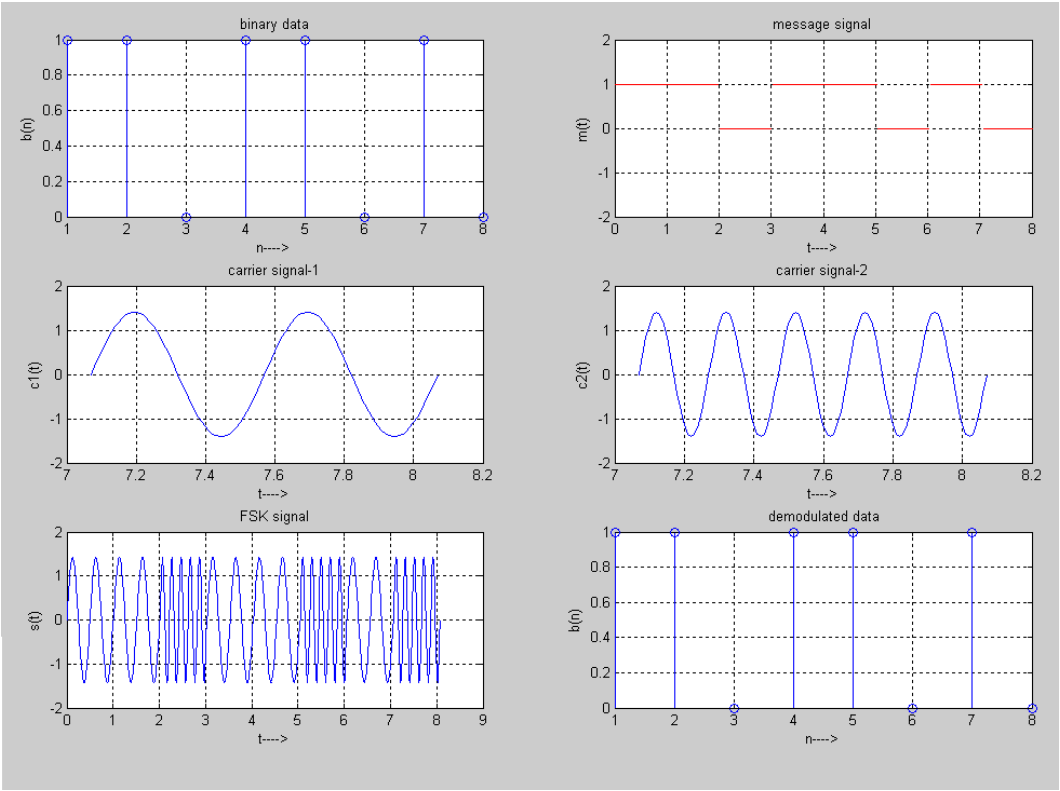


Fig :Matlab Simulation of FSK

```
else
mm(i+1)=c1(i+1);end
end
subplot(5,1,4)
plot(t,mm)
xlabel('Time')
ylabel('Amplitude')
title('Modulated Wave') for i=0:1000
if mm(i+1)==c2(i+1)
dm(i+1)=m(i+1);else
dm(i+1)=m(i+1); end
end subplot(5,1,5)
plot(t,dm)
xlabel('Time')
ylabel('Amplitude')
title('Demodulated signal')
```

PROCEDURE:

1. Run MATLAB

2. Open a new script file

3. Write the code for FSK technique.

4. Run the code for execution and obtain the necessary results

RESULT:

Thus the Frequency Shift Keying Modulation & Demodulation generation schemes were simulatedusing MATLAB.

DIFFERENTIAL PHASE SHIFT KEYING USING MATLAB

AIM:

To write a program to perform DPSK using MATLAB and to obtain its output waveforms.

APPARATUS REQUIRED:

Personal computer, MATLAB R2016a

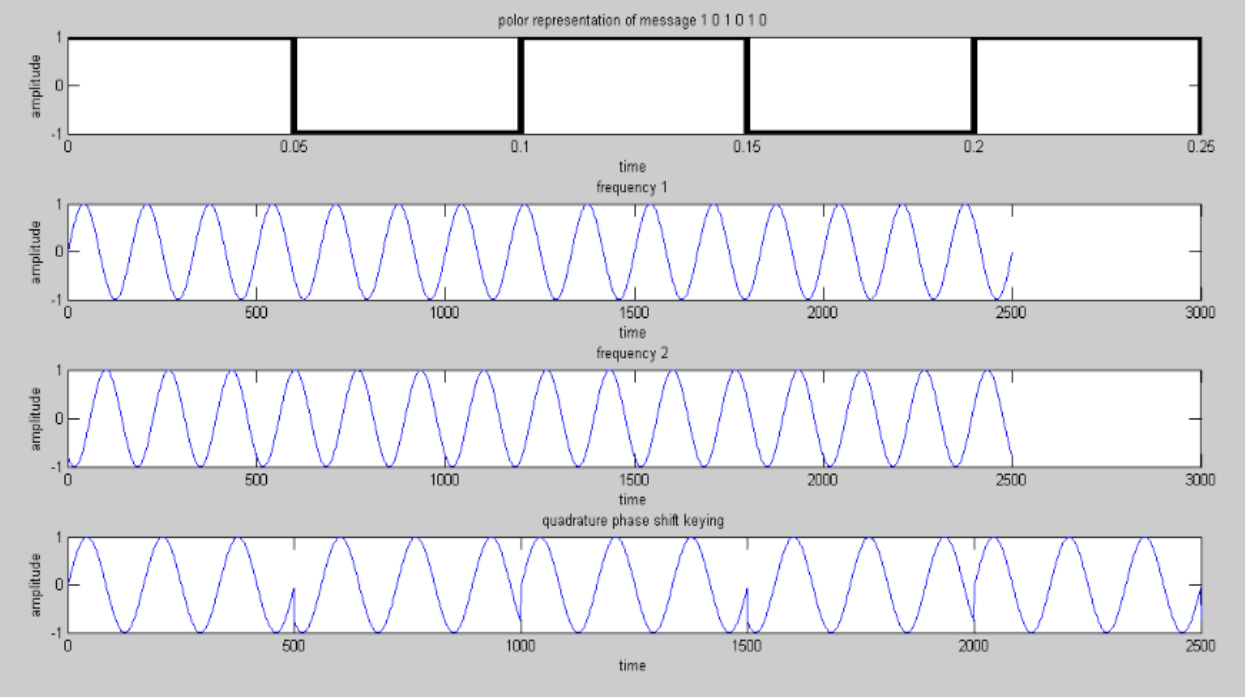
PROCEDURE:

1. Click start, select All programs, select MATLAB R2016a
2. Click file, select new M-file
3. Type the program
4. Save it with extension.m
5. Click RUN to obtain the output waveform

PROGRAM:

```
clc;
clear all;
rng default
M = 6; % Alphabet size
dataIn = randi([0 M-1],1011,1); % Random message
txSig = dpskmod(dataIn,M); % Modulate
rxSig = txSig*exp(2i*pi*rand());
dataOut = dpskdemod(rxSig,M);
errs = symerr(dataIn,dataOut)
errs = symerr(dataIn(2:end),dataIn(2:end))
figure
subplot(2,2,1)
plot(dataIn)
title(&#39;DATA&#39;)
subplot(2,2,2)
plot(txSig)
title(&#39;DPSK SIGNAL&#39;)
subplot(2,2,3)
plot(rxSig)
title(&#39;Received DPSK&#39;)
subplot(2,2,4)
plot(dataOut)
title(&#39;DATA RECEIVED&#39;)
```

SIMULATION WAVEFORM:



PROCEDURE:

1. Run MATLAB
2. Open a new script file
3. Write the code for DPSK technique.
4. Run the code for execution and obtain the necessary results

RESULT:

Thus the Differential Phase Shift Keying generation schemes were simulated using MATLAB.